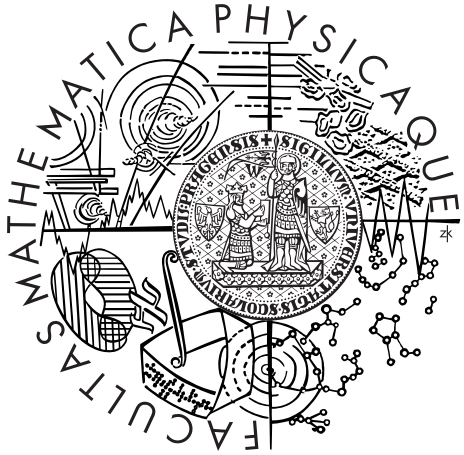
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Filip Děchtěrenko

Modelling eye movements during Multiple Object Tracking

Department of Software and Computer Science Education

Supervisor of the master thesis: Mgr. Jiří Lukavský, Ph.D.
Study programme: Informatics
Specialization: ITI

Prague 2012

I'd like to thank my supervisor Mgr. Jiří Lukavský Ph.D. for his gentle approach and valuable support. He kindly allowed me to access all equipment every time I needed it and he always helped me with any question I had. I'd also like to thank my family and closest friends for their support. Without them, I wouldn't be able to finish. I am deeply grateful to all my friends who participated on my experiments. Among many others, I'd like to thank especially Mgr. Cyril Brom Ph.D. for his highly valuable remarks to the content of my thesis and Silvie Mitlenerová for her help with language corrections.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Název práce: Modelování očních pohybů při sledování více objektů

Autor: Filip Děchtěrenko

Katedra: Kabinet software a výuky informatiky

Vedoucí diplomové práce: Mgr. Jiří Lukavský, Ph.D., PsÚ AV ČR, v.v.i

Abstrakt: Lidé musí sledovat v každodenních situacích více objektů zároveň (např. řízení automobilu nebo kolektivní sporty). Sledování více objektů (MOT) věrohodně simuluje sledování více objektů v laboratorních podmínkách. Když sledujeme cíle v úloze s mnoha dalšími objekty ve scéně, stává se obtížné rozlišit objekty na periférii (crowding). Přestože sledování by mohlo být prováděno pouze pomocí pozornosti, je zajímavá otázka, jakým způsobem lidé plánují své oční pohyby při sledování.
V naší studii jsme provedli MOT experiment, ve kterém jsme účastníkům předložili opakovaně několik úloh s proměnlivým počtem distraktorů, nahrávali jsme oční pohyby a měřili jsme konzistenci očních pohybů pomocí Normalized scanpath saliency (NSS) metriky. Vytvořili jsme několik analytických strategií, které se vyhýbají crowdingu a porovnali jsme je s očními daty. Kromě analytických modelů jsme trénovali neuronové sítě na předpovídání očních pohybů v MOT úlohách. Výkon navrhovaných modelů a neuronových sítí jsme vyhodnocovali na datech z nového MOT experimentu. Analytické modely vysvětlovaly variabilitu očních pohybů dobře (výsledky jsou srovnatelné s intraindividuálními rozdíly); předpovědi založené na neuronových sítích byly méně úspěšné.

Klíčová slova: modely, oční pohyby, sledování více objektů, neuronové sítě

Title: Modelling eye movements during Multiple Object Tracking

Author: Bc. Filip Děchtěrenko

Department: Department of Software and Computer Science Education

Supervisor: Mgr. Jiří Lukavský, Ph.D., Institute of Psychology, Academy of Sciences of the Czech Republic

Abstract: In everyday situations people have to track several objects at once (e.g. driving or collective sports). Multiple object tracking paradigm (MOT) plausibly simulate tracking several targets in laboratory conditions. When we track targets in tasks with many other objects in scene, it becomes difficult to discriminate objects in periphery (crowding). Although tracking could be done only using attention, it is interesting question how humans plan their eye movements during tracking.
In our study, we conducted a MOT experiment in which we presented participants repeatedly several trials with varied number of distractors, we recorded eye movements and we measured consistency of eye movements using Normalized scanpath saliency (NSS) metric. We created several analytical strategies employing crowding avoidance and compared them with eye data. Beside analytical models, we trained neural networks to predict eye movements in MOT trial. The performance of the proposed models and neuron networks was evaluated in a new MOT experiment. The analytical models explained variability of eye movements well (results comparable to intraindividual noise in the data); predictions based on neural networks were less successful.

Keywords: models, eye movements, multiple object tracking, neural networks

Contents

Introduction

In our everyday lives, we are surrounded by information which we have to process in order to survive. Because our cognitive skills are strongly limited, we have regulatory mechanism called attention; it filters stimuli which will reach consciousness while others remain in unconsciousness or are not processed at all. Attention has been studied for more than 70 years and one of the important findings is our ability to divide it between several stimuli. Pylyshyn and Storm (1988) developed multiple object tracking paradigm (MOT) for studying divided attention and since then, many interesting findings have been discovered. In typical MOT task participants see a set of objects and they have to track subset of them during motion, while other objects (called distractors) make tracking harder. In variants of MOT with large number of distractors phenomenon crowding occurs frequently. Crowding is defined as deleterious influence of nearby contours on visual discrimination (Levi, 2008) and is closely related to our capabilities in visual cognition. Although eye movements are not necessary for successful tracking, they provide us another source of information about tracking strategy and we believe that better understanding of eye movements can help us understand divided attention.

Because there is none Czech literature concerning presented subfield of vision, we propose translations of crucial terms (Attachment 1).

In our study we tried to determine if crowding in MOT influences eye movements. We prepared experiment in which we presented participants trials repeatedly while varying number of distractors. Eye movements were recorded during experiment using video based eye tracker and we compared consistency of those eye trajectories in repeated viewing of same trials. Consistency of eye trajectories was computed using Normalized scanpath saliency (NSS) for dynamic tasks (Dorr, Martinetz, Gegenfurtner, & Barth, 2010). NSS computes fixation map for several trials and consistency of new trial is then evaluated using this map. Our second goal was to develop several strategies which would explain variability of eye movements. There have been only several studies on eye movements strategies in MOT so far (Fehd & Seiffert, 2008; Zelinsky & Neider, 2008) but they did not take influence of distractors in account. We developed several analytical strategies which tried to minimize crowding during tracking and compared them with strategies from other studies. Strategies were compared using NSS metric similarly as in the experiment. In last part of our study, we tried to train neural networks to predict eye movements from positions of objects. Because eye data were noisy, we had to develop several optimization to reduce noise in order to successful train of neural networks. To validate our models, we replicated simplified version of the experiment and tested our models on data from this experiment.

We hope this study could serve as an introductory article to interesting field of modelling cognitive processes during visual perception from computer science perspective. Computer science and cognitive psychology could benefit from each other. Computer science may be a way for verifying crucial claims about human perception and findings from cognitive psychology could help us improve artificial intelligence.

1. Attention

The world around us is full of information, and only some of it is crucial for our functioning in our everyday lives. Our brain needs to process all relevant information, and react to it accordingly. To ensure that mind is not flooded with all kinds of useless data, we have a process called *attention*. Attention can be defined as the ability to focus selectively on specific stimulus while suppressing others. Attention is often related to the question of consciousness, and intensive research of attention may help us to understand what it is to be.

1.1 History of research on attention

Research on attention began during World War II, when it was discovered that people can not focus on several stimuli at once. Welford (1952) conducted an experiment where he presented subjects two subsequent signals separated only by hundreds of milliseconds. He found out that their reaction time to the second stimulus was reduced when the interval between stimuli was shorter than critical value. He called this interval *psychological refactory period*, and hypothesized that mind can start processing second stimulus after processing first one. Information thus can not be processed in parallel, and there have to exist some sort of *bottleneck* which filters amount of information.

1.1.1 Search for the bottleneck

First attentional research preferred auditory perception over visual because during visual perception unexpected visual stimuli can occur.

Dichotic listening and coctail party effect

Early research of attention tried to find where exactly is bottleneck located (Styles, 2006). Attentional research used *dichotic listening paradigm*. In dichotic listening task, subject is wearing headphones, different pieces of information are presented into his left and right ear, and he is told to attend to one of them. Presented stimuli are varied in order to determine what is consciously processed and what is not. An interesting phenomenon was described by Cherry (1953) while he was studying speech recognition. He noticed that people were able to selectively attend to one auditory input while filtering out other stimuli. It was named coctail party effect and many interesting questions have emerged (Bronkhorst, 2000).

Selective attention

There were several important selective attention theories trying to explain placement of bottleneck. They mainly differ in placement of filter during sensory processing. Broadbent (1952) presented *early selection theory* (sometimes known as Broadbent filter theory), based on his findings that people almost filter out information from one ear when they attend to the another one. According to early selection theory, all information is placed into sensory buffer, where it is

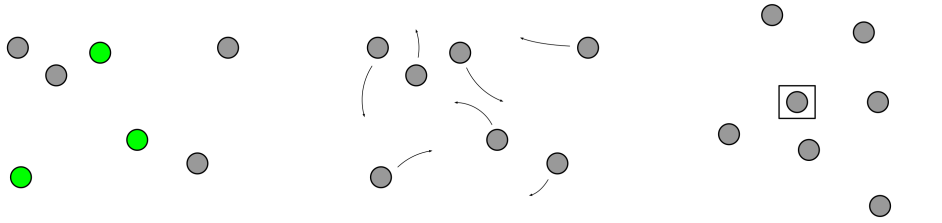
processed in parallel. All stimuli in buffer are evaluated and only those with specific physical attributes (as intensity of input sound, frequency, etc.) are selected for processing. Selected stimuli are then semantically processed while others are not processed at all (all-or-nothing approach). This theory did not correspond with findings that some words like subject's own name are captured even when they come from unattended source (Moray, 1959). Another filter placement was proposed by Deutsch and Deutsch (1963) in their *late selection theory*. They propose in contrast to Broadbent that all stimuli in buffer are semantically evaluated and those with the highest importance are consciously perceived while others are not. Filter is thus placed later in the process. Third theory which tries to find a location of bottleneck was theory of Treisman (1960). She placed filter in early stages of stimuli processing similar to Broadbent, but stimuli which were not salient enough are not thrown away, but they are passed for serial processing with weakened strength. This explains, why important stimuli like subject name reach consciousness although it comes from unattended source. Modern theories of selective attention work with resource models and are out of scope of this work. See Driver (2001); Navon and Miller (2002) for more information.

1.1.2 Metaphors for visual attention

In auditory processing, subjects can attend to physical traits of stimuli like frequency or wavelength, but similar discrimination was missing for visual stimuli. Simultaneously with placement of bottleneck metaphors of visual attention changed. One of first metaphors presented attention like *spotlight* which moved around visual field (Norman, 1968). Information from attentional spotlight is processed while everything outside is not processed at all (or it is processed with a weakened strength). Radius of spotlight is constant and could lead to processing of unimportant stimuli (if we move spotlight from some big object to a small one). As an alternative to spotlight with constant size, Eriksen and St James (1986) proposed metaphor of attention as *zoom lens*. According to zoom lens metaphor, subject can change radius of attended location but size of attended radius is inversely proportional to quality of information processing.
Both metaphors assumed that attention can be only focused on the continuous area. Awh and Pashler (2000) conducted an experiment when subject were told to attend to two distant places. Then cues were presented to random places on the screen and the result shown that reaction times (RT) were lower only at attended places and not in space between two places. Attention thus can be divided into several places, and we will refer to it as *distributed attention*.

1.1.3 Inhibition of return

We can move our attention voluntary from one place to another, but if some place is visibly *cued*, we will move our attention reflexive to that spot. After this reflexive movement, other attentional movements to the cued place will become inhibited for short period of time. This phenomenon is called *inhibition of return*. When we move our attention to cued place, this place becomes tagged and visual system does not need to return to that place. This is probably closely related to efficient visual search (Styles, 2006).



(a) Cue phase (b) Move phase (c) Query phase

Figure 1.1: Multiple object tracking task

1.1.4 Object based attention

With dividable attention a question arises if we attend to two unconnected places or if we attend to some complex structure. Duncan (1984) conducted an experiment where he showed participant a rectangle with a small gap and dotted line. Then they were asked two questions about attributes of those objects and it turned out that is it easier to make judgments about features belonging to the same object than judgments concerning features from different objects. Duncan proposed that our attention is object-based and if we determine attributes from two different objects, we have to lift our focus from one object, move it to another one and process informations from second object.

1.2 Multiple object tracking

One of widely used paradigms for study of distributed attention is *multiple object tracking* paradigm (MOT), which was introduced by Pylyshyn & Storm (1988). In typical MOT task (Figure 1.1), subject is presented n objects (e.g. $n = 8$), and objects x_i (where $i = 1 \dots m < n$) are highlighted (they flash or change color) for several seconds (1.1a). Then all objects return to their original state and move randomly around screen (1.1b). After several seconds motion stops, and subject is asked if one or more objects belonged to m (1.1c). We will denote x_i as *targets* and x_j $(m < j \leq n)$ as *distractors*. In some variations of MOT, light flashes on some objects during motion and later subject is queried about those objects. This technique is called *cueing* and it helps us determine current attentional distribution.

MOT is very good paradigm for studying distributed attention, because unlike other attentional paradigm subject have to keep attention on the targets through whole motion because they are indistinguishable during motion. Another great advantage of MOT is its naturalistic interpretation. Subjects encounter MOT in everyday situations e.g. watching several players during collective ball game or watching children at playground. Important real-life application of MOT task is airplanes traffic control where extreme tracking accuracy is required.

1.2.1 Facts about MOT

Multiple object tracking has been widely studied for almost 25 years and many factors influencing tracking have been discovered. On average, people can track 4 targets out of 8. We will denote this configuration as 4:4 (in general m:(n-m)).

When we increase number of distractors (m), average success rate of tracking decreases (Alvarez & Franconeri, 2007). The increase of movement speed of objects also reduces tracking accuracy (Verstraten, Cavanagh, & Labianca, 2000). The accuracy is also reduced when tracking time increases (Oksama & Hyönä, 2004). Objects typically do not move around whole screen because monitor borders can be interpreted as another object and interfere with tracking mechanism. Instead, objects move in a smaller frame which size affects tracking accuracy (smaller frame leads to reduced accuracy). If we use very small frames we will non be able to track at all. This is probably caused by the lower limit of attentional resolution (Intriligator & Cavanagh, 2001). Complexity of objects plays important role in tracking. Usually simple circles are used, but when we use 4 lines instead and subjects have to track one part of line, tracking accuracy is reduced. It seems that visual system have to have clear boundaries between targets and distractors (Scholl, Pylyshyn, & Feldman, 2001).
Alvarez, Horowitz, Arsenio, DiMase, and Wolfe (2005) showed that if targets disappear for a short period of time and then reappear on same positions, it leads to better accuracy than if targets moved while they were invisible. This is a controversial finding, because it claims that object movement is probably not predicted during tracking.

1.2.2 MOT models

There have been several attempts to explain mechanism behind tracking. First two theories tried to explain tracking theories without divided attention and worked only with one attentional spotlight, the other two use more sophisticated assumptions.

- *Switching model* (Pylyshyn & Storm, 1988) assumes that people rapidly switch their attention between targets' last locations and if they are still in focus, they update indexes of their new locations. Attention have to switch between targets fast enough otherwise targets can move too far or another object can move into target's location. This model (in its most simple form without any form of movement prediction) was later shown as not biologically plausible. Subjects were able to track several targets which were so far from each other that attention have to travel on the screen faster than any biological data support.

- According to *Grouping model* (Yantis, 1992) people create one coherent object from targets (targets are vertexes of polygon) which is tracked with attention. This model has good results, if subject is explicitly told to form an object from target, performance usually improves.

- *Multifocal attention model* (Cavanagh & Alvarez, 2005) assumes that attention can be divided. Visual system assigns one focus of attention to each target and moves together with them. When objects stop moving, each attention is focused on same targets as before. Tracking capacity of four targets corresponds to four different attentional foci. The identity of object is not maintained, so visual system is able only to determine if it was member of m.

- In *FINST model* (Pylyshyn & Storm, 1988), visual system assigns indexes to the targets before movement. Indexes can be pictured as fingers pointing at specific features of object. Those pointers move with objects. Main distinction between FINST and multifocal attention is in role of attention in this process. In FINST, indexes are assigned in preattentional phase, so attention is not needed for tracking.

1.3 Crowding

Crowding is visual phenomenon which limits our perceptual cognition. Crowding can be defined as deleterious influence of nearby contours on visual discrimination (Levi, 2008). We can see crowding effect on Figure 1.2. If we fix our eyes on central cross, we are able to identify letter **A** on the left, but we can not identify letter **A** on the right, because it is crowded by letters **S** and **H**. We will call surrounding objects *flankers*. Crowding is responsible for lack of acuity on periphery and probably corresponds to critical spacing in reading (Pelli et al., 2007).

Figure 1.2: Example of crowding effect. While fixating on cross, letter **A** on the left can be identified, but letter **A** flanked by letters **S** and **H** can not.

1.3.1 Facts about crowding

Crowding is widely studied for almost 70 years and wide range of factors affecting crowding have been discovered. Main factor which determines if crowding occurs is ratio of spacing between target and flankers and eccentricity. Bouma (1970) experimentally found out that this relation is linear and crowding occurs when ratio of object spacing to eccentricity is approximately 1:2 (distance of objects were measured between centers of objects). This ratio is known as *Bouma's law*. Question arises, if crowding occurs on fovea as well, but according to Bouma's law, objects should be so close that visual system would interpret that as occlusion.
Flanker-target similarity affects crowding. If they are partially similar crowding effect is bigger but when target and flankers are the same, crowding does not occur (Pelli, Palomares, & Majaj, 2004). If target and flankers differ in color or in size, crowding is reduced (Kooi, Toet, Tripathy, & Levi, 1994). Crowding is asymmetric - if we have two letters next to each other, the further from the fovea will be easier to identify (Bouma, 1973). Crowding in upper part of visual field is stronger then in lower part (He, Cavanagh, & Intriligator, 1996). This could arise from evolution needs, because people were always more threatened from the land then from the air.

1.3.2 Crowding models

There are many models explaining the mechanism of crowding – from low-level anatomical models working with structure of eye to high-level attentional models

(see Levi, 2008). All of them propose that visual information is processed in two phases:

1. Visual system detects simple object features in visual field.

2. Features are integrated into coherent objects. Visual field is covered with *integration fields* and if two or more features fall into one field, visual system binds them to same object. Then recognition of whole objects starts and in this phase crowding occurs, if features are binded incorrectly.

Size of integration fields is not constant. They are larger further to the periphery, while near fovea they are smaller. This correlates with fact that crowding occurs more on periphery.

1.4 Eye movements

Study of attention can tell us a lot about the process of acquiring information. Main problem with attentional experiments is lack of direct measurements where is attention focused and we can rely only on the indirect ones which measure how tracking accuracy is influenced by varying some parameters of the paradigm. One possibility, how to partially determine where subject is focusing his attention is to measure eye movements. Attention can be focused on different place to where eyes look (Posner, Snyder, & Davidson, 1980), but in complex information processing tasks, they are probably closely related (Rayner, 1998).
Eye movements have been studied for more than 100 years and we can divide research into three eras. First era (from 1897 to 1920s) was mainly focused on eye movements in reading and many important facts have been discovered. Second era of research (1920s-1970s) started to infer cognitive processes via eye movements, but it was strictly limited with current technology. Third era (since 1970s) started with technological advance when more accurate measurement systems were available.

1.4.1 Eye trackers

There are three types of eye trackers used for recording eye movements (Rayner, 1998):

- Contact eye trackers – eye trackers of this type attach special contact lens to the eye and assume that this lens will not slip during rotations of the eye. There can be embedded mirror inside of contact lens and this tight connection provides extremely sensitive eye tracking capabilities.

- Non-contact eye trackers – these eye trackers measure eye movements using cameras or some other optical sensors. Video based eye trackers compute eye coordinates from center of pupil and corneal reflection. Optical methods are non-invasive and often used for gaze tracking.

- Tracking using electric potential measurement – eye trackers from third group measure electric potential with electrodes placed around eye. When eye moves it changes electrical potential field and from those changes are eye

coordinates calculated. Measuring of electric potential is very good method for detecting and measuring saccades and blinks, but it has problem with slow eye movements.

Although there has been discussion about measuring and evaluating eye movements (McConkie, 1981), no measurements standards have been adopted. However, even with the lack of measurements standards, many studies regarding eye movements have been successfully replicated (Rayner, 1998).

1.4.2 Visual angle

Eye consists of lens which project objects to retina. If we want to measure some phenomena of visual system, object size in metric unit would be ambiguous, because if we have two objects of size s_1 and $s_2 = 2s_1$ in distances d_1 and $d_2 = 2d_1$, they would have same size on the retina ($\frac{s_1}{d_1} = \frac{s_2}{d_2}$). In research of visual perception, degrees as unit are used instead to capture the ratio between real object size and its distance. We can convert real object size to degrees using formula

$$\alpha = 2 \arctan \frac{s}{2d}$$

where s is real object size, d is distance of object from eye and α is object angular size.

1.4.3 Types of eye movements and visual acuity

There are four basic eye movement types: saccades, smooth pursuit, vergence movements and vestibulo-ocular movements. *Saccades* (fast movements of the eye changing location where the eyes look) are most important for processing of information. Intervals between saccades when eyes are relatively still are called *fixations*. Visual system does not acquire any information during saccades, this phenomenon is called *saccadic suppression*. It is still an open question, whether cognitive process are suppressed during saccades as well (see Rayner, 1998). Saccades and fixations are measured in milliseconds and their length depends on task:

- saccades - from 30 ms in reading tasks to 50 ms in scene perception
- fixations - from 225 ms in silent reading task to 330 ms in scene perception

Information is processed only during fixations and efficiency of processing depends on area of visual field. Visual field can be divided into three regions with different acuity and efficiency of processing information:

- *foveal* - central circle with diameter 2°, best efficiency
- *parafoveal* - annulus with radius from 2° to 5°
- *peripheral* - rest of visual field, worst efficiency of processing

When some stimulus lies in parafoveal or peripheral region visual system could make a saccade to bring object near fovea, if characteristic of stimulus requires so. We can divide visual field by different criterion to regions (Sanders, 1967) where

- stimulus can be identified without eye movement
- stimulus can be identified only with eye movement
- stimulus can be identified only with head movement

Even during fixations eyes move a little. We can distinguish three types of those small movements: nystagmus, drifts and microsaccades. Those movements are probably results of imperfect control of occulomotor system. In majority of experiments those movements are considered as noise.
Another type of eye movement important for MOT tasks is *smooth pursuit*. Smooth pursuit is much more slower then saccades and information is not suppressed during this kind of movement.

1.4.4 Comparing eye trajectories

Eye trajectories can be described as sequence of tuples $(x_i, y_i)_{i=1}^{t}$, where x_i and y_i are coordinates of eye in time i and t is total time of eye movements. We will call this trajectory of eye movements in time as *scanpath*. This term was first used by Noton and Stark (1971) in their controversial theory. They assumed that when we saw a new object, we store succession of fixations into memory and then later at recognition phase we simply follow the stored scanpath. This theory is now obsolete, but term scanpath is still used. Synonym to scanpath is *scan pattern* which is not related to theory of Noton and Stark.
It is a quite difficult task to compare scanpaths and there is no obvious metric for comparison. If two trajectories are twice as far, are they twice as different? What if two trajectories are similar, but just shifted in time, how different they should be? *Good similarity metric* should meet following criteria (Dorr et al., 2010):

- It should be resistant to large outliers.
- Trajectories, when all but one of the subjects look at location A and one subject looks at some other location B, should be more similar then if half of subjects look on the A and the other half on the B.
- There should be no hard threshold on similarity, because of inaccurate spatiotemporal measurements of eye trackers.
- Its values should have intuitive meaning.

We would like to present some approaches to comparing similarity of scanpaths.

Clustering algorithms

One approach to comparing similarity are clustering algorithms. Clustering algorithms divide fixations into the clusters and then compute for each trajectory percentage, how much of trajectory fixations falls into the all but one clusters. Commonly used clustering algorithms are EM and k-means. See Duda, Hart, and Stork (2001) for description of those algorithms. Clustering algorithms are resistant to outliers and they are intuitive, however they use fixed threshold value for determining whether fixation falls into the cluster. Santella and DeCarlo (2004) presented smoothing of thresholds, but it makes scaling of the cluster unpredictable, so even two distant fixations can be classified as similar.

Editing algorithms

Another method for measurement of similarity is editing algorithms. These algorithms assign letters to the regions of space and then convert scanpaths to corresponding strings of those letters. We can compare similarity of two strings by string editing algorithm which sums penalties for insertion, deletion or mismatch of letters. Problem with editing algorithm is that they need to have a priori specified regions of interest for letter penalty table. Editing algorithms usually did not consider order of fixations, but there exist modifications for cases where order of fixations matters (Clauss, Bayerl, & Neumann, 2004).

Fixation map

One of noise resistant methods operates with fixation maps (noise resistant means that there can be some artifacts in eye trajectories). Fixation maps are created by additive superpositions of Gaussians which are centered at each fixation location. Summing of fixations from different scanpaths produces smooth value map which does not suffer from large outliers. Fixation map similarity metric is often defined as $\sum_{i,j} (f_x(i,j) - f_y(i,j))^2$ where $f_x(i,j)$ and $f_y(i,j)$ are values from fixation maps x and y at position (i,j). Fixation maps are good metric for static scenes (Dorr et al., 2010).

Kullback-Leiber Divergence

Kullback-Leiber Divergence is a robust method based on information theory. It was introduced by Rajashekar, Cormack, and Bovik (2004) and improved by Tatler, Baddeley, and Gilchrist (2005). KLD specifies information provided from one distribution given knowledge about another distribution. KLD is good similarity measure, but it is not intuitive - identical scanpaths have zero KLD value and we do not have straightforward interpretation for non identical scanpaths.

1.5 Conclusion

Visual perception is an interesting field of study. We have introduced basic information about attention and presented paradigm multiple object tracking for research of distributed attention. We have described phenomenon crowding which occurs during visual perception. Finally, we have introduced some basic facts about eye movements and we did a short review of techniques for comparing eye trajectories. In out study, we decided to use Normalized scanpath saliency metric (NSS) which meets all criteria for good similarity metric. We will present this metric later in more thorough form.

2. Machine learning

In this chapter we would like to review several concepts about machine learning in general and especially neural networks, which we will used later in the study. Machine learning is a subfield of computer science which focuses on techniques which can be used for learning some pattern in data. Machine learning algorithms can be divided into four groups: *supervised learning*, *unsupervised learning*, *semi-supervised learning* and *reinforcement learning*. For supervised learning techniques, we have training inputs and corresponding outputs. Supervised learning can be used for *prediction* - program tries to predict value for given inputs or *classification* - program for given sample determines which class it belongs to.

2.1 Learning strategies

In machine learning in general, we assume we have set of training data T. This set is divided into three subsets:

- Training set - this set will be used for training model
- Validation set - this set will be used for cross validating learned model
- Testing set - this set will be used for performance testing of model

Typically, we divide T in ratio 0.7:0.15:0:15 (Testing:Validation:Test) to ensure best performance. Sometimes validation set is omitted but it can lead to *overfitting*. Overfitting means that algorithm learned training data too well and does not generalize well on other inputs. If we stop training too soon, it could lead to *underfitting* which means that algorithm did not learn data well.

2.2 Artificial neural networks

Artificial neural networks (ANN) are mathematical models used for machine learning. Artificial neural networks are inspired by neural connections in human brain. The first concept of artificial neuron was described by McCulloch and Pitts (1943). Another important father of ANN was Donald Hebb (Haykin, 1999) who extended concept of McCulloch and Pitts with rule currently known as the Hebb Rule. In 1958, Frank Rosenblatt showed limitation of perceptron in inability of learning XOR function. In the 1980's research of ANN was founded and they have been widely used since then. ANN have many applications in computer science, biology, economics or psychology.

2.3 Multi-layer perceptron network

Best known (and probably most used) are *multi-layer perceptron networks*. Multilayer perceptron (sometimes referred as feedforward network) consists of *input*

layer, one or more *hidden layer* and *output layer*. Signal is propagated from inputs to outputs (feed forward) using *activation function*. MLP are generalization of simple perceptron and can be used for classification or regression.

2.3.1 Structure of network

In general, we can visualize neural network as oriented weighted graph with specified input and output subsets of vertices. Typical example of neural network can be seen on Figure 2.1.

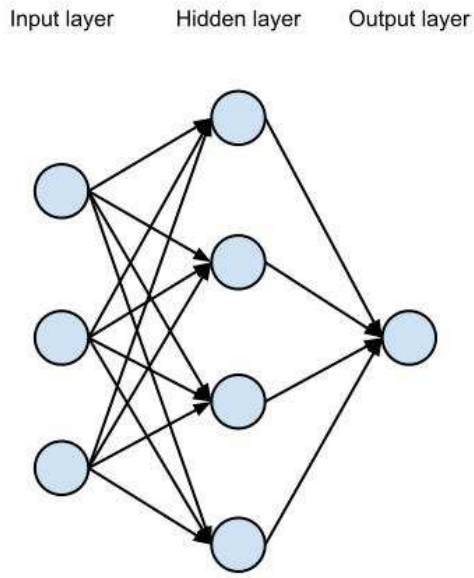


Figure 2.1: Example of neural network with one hidden layer.

General neural network does not have any constraint on edges, so there can be recurrent edges or cycles. *Multi-layer perceptron network* is a special case of ANN, where following conditions hold:

- Graph is acyclic.
- Set of neurons N can be divided into k subsets $N_1, \ldots, N_k$, where each two subsets are disjunctive. Those subsets are called *layers*.
- There are all connections between neurons from two subsequent layers are there no connection between neurons in one layer or between two non-subsequent layers.

We will denote N_1 as *input layer*, N_k as *output layer* and $N_2, \ldots N_k - 1$ as *hidden layers*. With these conditions, we can define MLP network (Trenn, 2008) as quintuple

$$M = (k, \vec{n}, \mathbf{W}, \vec{\sigma}, \vec{\vartheta}),$$

where

- k is number of hidden layers

- $\vec{n} = \{n_0, n_1, \ldots, n_k\}$ is vector of number of neurons in hidden layers
- $\mathbf{W} = \{\mathbf{W^1}, \ldots \mathbf{W^k}\}$ is vector of matrices of weights between layers
- $\vec{\sigma} = (\sigma_1, \ldots, \sigma_k)$ is vector of *activation functions* for every layer
- $\vec{\vartheta} = \{\vartheta_1, \ldots, \vartheta_k\}$ is vector of *biases*

Values of weight matrices should be chosen uniformly from interval $\langle -\alpha, \alpha \rangle$ for some α with mean value zero.

2.3.2 Learning

If we are using neural networks, we need to train them to correctly compute values to input data. We will assume that set of input data T is in format $T = \{(x_i, d_i)\}_{i=1}^n$, where d_i is output for input x_i and n is count of samples. We present neural network one sample at time and network adapts for this sample. Then we repeat this process, until network is trained. We try to minimize error for training set

$$E = \frac{1}{n} \sum_p \sum_j (y_{j,p} - d_{j,p})^2,$$

where n denotes number of samples, y_j, p denotes output value of j-th neuron in output layer for p-th training sample and $d_{j,p}$ denotes desired output of p-th sample d_p. This error is in literature often called *mean squared error*. Sometimes we add to error function term $\frac{\lambda}{m} \sum_{i,j} w_{i,j}^2$, where $w_{i,j}$ are all weights between layers, m is number of all weights and λ is parameter regulate contribution of this term to overall error. This term is used for preventing overfitting and this technique is called regularization.
Simple algorithm for training neural network is called *backpropagation*. This algorithm computes gradient of the error of the network and adapts weight accordingly. For correct behavior of this algorithm, activation function has to be differentiable (this holds for logical sigmoid). Training of neural network using backpropagation for one sample consists of two steps.

- In the first step, we present network an input vector $\vec{x}$. Then we propagate this information from input layer to output layer through hidden layers. To pass value from i-th layer to to $i+1$-th layer, we will use formula:

  $$y_j = \sigma_i(\sum_{j'} y_{j'} w_{j',j} + \vartheta_j),$$

  where y_j represents output of j-th neuron in $i+1$ layer, $w_{j',j} \in \mathbf{W^i}$ is weight between j'-th neuron in layer i and j-th neuron in layer $i+1$, ϑ_j is bias for j-th neuron and σ_i is an activation function for i-th layer. One of the most used activation functions is *logical sigmoid function* and is computed from formula

  $$\sigma(\xi) = \frac{1}{1 + e^{-\lambda \xi}}$$

  Argument ξ is called *potential* inspired from biological neural networks. Learning parameter λ changes steepness of sigmoid around zero. Another

possibility for activation function can be hyperbolic tangent sigmoid or linear function (linear function is often used as activation function in output layer, if inputs are normalized). Hyperbolic tangent sigmoid is similar to logical sigmoid and it is often used for better performance.

- In second step, we will compute error of network for input x_i using formula

$$\sum_j (y_j - d_j)^2$$

and adapts weight function recursively from output layer to input layer (that is why is this algorithm called backpropagation). For MLP with sigmoid activation functions, weights are adapted using following rule:

$$w_{j',j} := w_{j',j} + \alpha \delta_j y_{j'}$$

where

$$\delta_j = \begin{cases} (d_j - y_j)\lambda y_j(1 - y_j), & \text{for output neuron.} \\ (\sum_k \delta_k w_{j,k})\lambda y_j(1 - y_j), & \text{otherwise.} \end{cases}$$

and α is parameter controlling speed of convergence.

For proof see (Haykin, 1999, p. 183)
Training neural network with sigmoid function is NP-hard, but in real life situations, neural networks converge to optimum quite fast.
There are many other training algorithms, for example: Gradient descent with momentum (Rojas, 1996, p. 186), stochastic gradient descent (Amari, 1993) or Levenberg-Marquardt backpropagation (Hagan & Menhaj, 1994). Levenberg-Marquardt backpropagation tries to find numerical solution to the error function minimization using Jacobian matrix. Detailed description of this algorithm is beyond scope of this study.

2.3.3 Function approximation

One of great advantages of MLP are their universal approximation capabilities (Šíma & Neruda, 1996). If we have MLP with one hidden layer and activation function is continuous, nonconstant and bounded on set $X \subset \mathbb{R}^n$, we can approximate every function from $C(X)$ arbitrary close, if hidden layer has enough neurons. Here $C(X)$ denotes metric space of all continuous functions over compact space X with supremum norm

$$\|f\| = \sup_{x \in X} |f(x)|$$

For proof see Šíma and Neruda (1996, p. 335).
This is very useful theorem. We do not need to create MLP with more than one hidden layer, because we can achieve same approximation with more neurons in hidden layer.

3. CrowdMOT Experiment

In first MOT experiments, research was focused on performance during tracking. Pylyshyn and Storm (1988) were interested only on attentional tracking so they measured eye movements and excluded trials in which eyes move too much. Twenty years later situation begin to change and question arises, what eye movements can tell us about attention. We designed an experiment in which we try to find answers to questions concerning eye movements in information rich tasks such as multiple object tracking with crowded displays.

3.1 Introduction

Most research of MOT tracking used classical ratio 4:4 of targets and distractors. We assumed that if we increase number of distractors, task will become more demanding on processing of information and eye movements and attentional focus would be more related. We presented subjects with some tracks repeatedly and we measured consistency of their eye movements. There are several ways how to measure eye movements, we used normalized scanpath saliency (NSS) metric which is good measure for comparing consistency of trajectories. On scenes with more objects, crowding would occur more and we wanted to find out, if it would influence consistency of eye movements. We wanted to examine direct effect of crowding on tracking, so we created special type of trial, in which presented number of distractors has varied across blocks. If crowding has no effect on tracking, we should got similar consistency as in repeatedly presented trials.
To ensure that tracking would be similarly difficult for all participants, we varied movement speed using staircase method described by Cornsweet (1962) modified for MOT tasks.
Our another goal was to find some analytical strategies which could explain variances in behavioral data from experiment.

3.2 Method

3.2.1 Participants

Ten subjects (4 males, 6 females) have participated in experiment. Mean age was 22.18 years. They all have participated voluntarily and all of them have normal or corrected to normal vision. They were naive to the purpose of the experiment and none of them have participated in multiple object tracking task before.

3.2.2 Apparatus

Experiment was programed in MATLAB with installed Psychtoolbox-3 (Brainard, 1997; Pelli, 1997; Kleiner, Brainard, & Pelli, 2007). Psychtoolbox-3 is set of MATLAB and GNU/Octave functions for vision research. Its main advantage is low-level approach which enables exact stimulus control. On the other hand MATLAB is robust high-level interpret language which makes process of creating experiment easier than coding it in low-level language only. Brief documentation

can be found in Attachment 2 and more thorough on the cd. Because we needed to have latency as low as possible, we presented experiment on operating system Ubuntu 10.10 with real time kernel. Experiment is able to run on Windows, but we will not get exact timing. We used 19" CRT monitor with resolution 1024 x 768 and frequency 85Hz. LCD monitor is not good for displaying data in experiments, because they produce shadow artifacts after movement. Participants have their head positioned on the chin rest 50 cm away from screen to ensure same visual angle.

Eye tracking

Participants have eye tracker positioned on their head during whole experiment and it recorded their eye movements. We used video-based eye tracker EyeLink II version 1.05, because it has highest resolution (noise limited at $< 0.01°$) and fastest data rate (up to 500 samples per second). EyeLink II is ideal for saccade analysis or smooth pursuit tasks such as multiple object tracking and it uses heuristics for detecting saccades and fixations. We worked with eye positions only because tracking in MOT is mostly done by combination of smooth pursuit and saccades and we were not interested in classification of types of eye movements. EyeLink II consists of an adjustable headband with attached cameras which scan pupils. There are four markers attached on monitor which are read by small camera on the headband. Those markers are used for computation position of the headband in reference to monitor. Main experimental computer was connected to another computer with a software to control eye tracker. During experiment, data were sent to this computer and when experiment finished, binary file with trajectories was sent back to the experimental computer.
To minimize error of measurements of eyes, eye tracker had to be calibrated for each participant and we had to make drift correction often to update coordinates for each trial, because headband can slide and then all coordinates would be incorrectly computed. We used 9 point calibration which ensures little tracking error while achieving this error should not be problem for most participants. During 9 point calibration, one dot was shown on the center of the screen and when participant looked on the dot, it moved to another place on the screen. Dot changed its position 9 times in total and coordinate system (reference for eye positions towards monitor) was computed. Calibration was followed by validation, in which dot moved around screen as in calibration and measured eye positions were compared to coordinate system. After calibration, the eye with lower error is selected for tracking. In drift correction, participant had to look on a dot in the center and eye tracker corrected coordinates to compensate for error between central dot and eye coordinates. Drift correction is very useful when correcting for slides of headband.
EyeLink II sends data every 4 ms (250Hz) and each data record of eye movements consists following information:

- x and y coordinates – coordinates were measured towards center of the screen; in pixels
- time stamp – time information were taken from internal eye tracker counter; in ms

- pupil size – radius of measured pupil; in arbitrary unit

3.2.3 Procedure

Experiment was divided into two parts: calibration and testing. Main purpose of calibration phase was to find out individual movement speed of objects for each participant in which he can track accurately in 4:12 task (medium difficulty). This speed adjustment should make classification of difficulty of trials consistent.

Stimuli

We used gray dots with radius 0.5°. In testing phase the number of presented dots varied in each trial from minimum ratio of targets and distractors 4:4 to maximum 4:20. We classified trials to three categories based on number of distractors in trial:

- easy trials – 4-8 distractors
- medium trials – 9-13 distractors
- hard trials – 14-20 distractors

Calibration part

Trials in calibration part consisted of cue phase, move phase and query phase. In cue phase targets were highlighted (they changed color to green) for 2 s and participant had to prepare for tracking. In move phase targets' color returned to gray, and all dots began to move. Objects were moving randomly for 8 s in rectangle $30° \times 30°$ and they bounced from borders of that rectangle (similar behavior as light reflection). Dots occluded each other, but in the last 5% percent of the trial they started to bounce to ensure that they would not stop in the same place because it could confuse participants. Each dot could change direction in each frame with probability 3%. In query phase all dots stopped moving and participants had to select targets with mouse. A dot changed color to yellow on click. If subject correctly selected all targets, speed for next trial was increased by $0.3°/s$, if he had one or more mistakes speed decreased by $0.6°/s$. There were 15 trials in total in calibration and trajectories for every trial were generated on the fly. Final speed from calibration part was used as movement speed in testing part. Eye movements were not measured in calibration phase. Number of targets selected correctly was shown after each trial.

Testing phase

Trials in testing part of experiment had the same cue phase and move phase, they differed in query phase. In query phase yellow square frame was shown around one of dots and participant had to respond if the dot belonged to the targets. Participants responded by left and right arrow (right arrow if dot was one of the targets, left if it was not). If participants responded correctly, color of queried dot changed to green, if incorrectly, color changed to red. Experiment consisted from 4 blocks, in each block there were 20 trials. There were three types of trials:

- *Repeating* – repeating trial was presented in each block once, so participant saw the same trial repeatedly for four times.
- *Changing* – in changing trials, one trajectory was generated for configuration $4 : 20$ and in each block, different subset of distractors was presented. Presented subsets were ordered by inclusion. Formally if N was set of distractors, for changing trials and $s_1 \subset s_2 \subset s_3 \subset s_4 \in N$ were presented subsets in each block, but they were presented in random order.
- *Random* – each random trial was presented only once for each subject. Random trials were used to mask the fact that some some trials were presented repeatedly (repeating trials and changing trials)

There were 3 different repeating trials for each difficulty category (3×3), 5 increasing trials and 6 random trials in each block, 20 trials total. Order of trials in each block was random and each subject had different set of trials. All trials were generated prior to experiment with movement speed $5°/s$ and because movement speed for participants could differ based on their performance in calibration phase, dots trajectories were interpolated to move with participant's speed.
Eye movements were recorded only during cue and move phase, we were interested in eye trajectories only during tracking. We calibrated eye tracker before each block and we did drift correction before each trial. Participants did not take off eye tracker between blocks. Experiment was administered by one experimenter and all participants heard the same instructions. Each participant was queried after experiment if they used some strategy for tracking and asked about their personal experiences of experiment.

3.3 Results

3.3.1 Calibration phase

It turned out that calibration phase was too hard and 8 participants ended at minimal speed 2°/s and 2 participants with speed 2.2°/s. Slow movement speed could lead to more eye movements because participants would not have lost as much information about dot positions during saccades because of saccadic suppression. Because participants responded by selecting targets, we had ensured that they successfully tracked all targets. It means that eye movements for medium difficulty trials should be correlated with strategy employed for successful tracking 4 targets.

3.3.2 Parsing data

Raw data from eye tracker were converted from binary format to ASCII format using utility software from EyeLink II. We sent additional messages into file with eye movements during experiment to identify which eye trajectories belong to which trial. However, EyeLink II sometimes did not send messages correctly so we were not able to identify cue or move phase in all cases. All missing cue phases and move phases were inspected by hand, because each trial can be missing from several reasons and sometimes we were able to add missing information into raw

data and to repair incorrectly added messages. Because EyeLink II used internal timer and sampling rate was 4 ms, trial starts could differ by 1-4. To ensure comparability of eye records from different trials we normalized all trials to length 7400 ms, so each trial contained 1850 samples.
Trial was 10 s long, so people sometimes blinked. We had to find and remove blinks by our heuristics. In general, eye blink are identified as fast vertical movement with decreasing pupil size. We discussed several criteria for finding blinks:

1. samples with y coordinates $> 15°$ below screen center.
2. samples with saccade speed $< s_l$ where s_l is experimentally stated critical saccade speed
3. samples with pupil size $< p_l$ where p_l is experimentally stated critical pupil size

Variant 1 has been shown as not a conclusive criterion because not all blinks were classified as vertical movements below the rectangle in which dots moved. We computed saccade speed for two consequent coordinates, but we found this criterion inconclusive as well. Variant 3 seemed as best criterion for finding blinks. We have inspected several eye trajectories and set critical value p_l for blinks as 75% of maximal pupil size in trial. This condition was good enough to find most blinks (see Discusion for better methods for finding blinks). We also discarded all samples where x and y coordinates were $> \pm 15°$ because it means that participant was distracted or tired and looked away from the rectangle, in which the dots have moved. Sometimes headband could slide a little (participant wrinkle his forehead). This was demonstrated as a "long look" outside rectangle. Because we wanted to compare trajectories, we needed only trajectories with minimum data removed. Trials with more than 10% of eye data removed were discarded as invalid. We can see some typical examples of eye trajectories on Figure 3.1

We have to discard 8% of all trials because of following errors:

- missing data – 17%; no data was recorded for this trial
- wrong size of trial – 50%; this error occured if some of messages separating phases were sent incorrectly
- too much blinks – 33%; trial were discarded because of too much blinks / too much looks outside of rectangle

Overall tracking accuracy was 91.1%, we did not have to exclude any participant because of low accuracy (lowest accuracy was 80%). Accuracy decreased with difficulty as we can see on Table 3.1. Difference in accuracy is significant ($F(2, 797) = 12.35, p < 0.001$) and decreases with increasing number of distractors. The observed accuracy is still higher than chance level.

3.3.3 Comparing trajectories

After preprocessing data, we compared trajectories using *Normalized scanpath saliency* (NSS) method. NSS was described by Peters, Iyer, Itti, and Koch (2005) for static scenes and modified by Dorr et al. (2010) for dynamic scenes. It meets

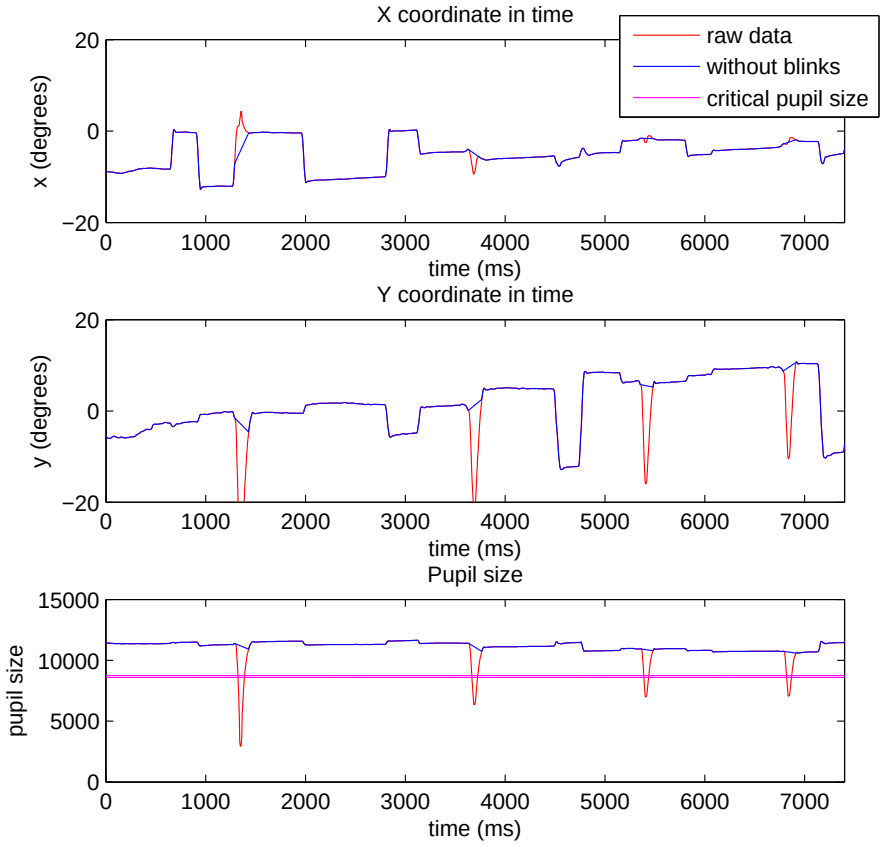


Figure 3.1: Typical plot of eye coordinates and pupil size in time. Red lines are unprocessed data, blue lines are data after removal of blinks and violet line corresponds to critical value for pupil size.

all criteria for good similarity metric and it s fast to compute. NSS computes sum of fixation maps for all trajectories but one and then normalizes this distribution to have unit standard deviation. From this scaled distribution we compute NSS value with intuitive meaning. NSS values around zero correspond to uncorrelated scanpaths, negative values correspond to highly dissimilar eye movements. We get high NSS value for trajectories which tends to be similar as summed fixation map. Formally:
Let $\vec{x_k} = (x, y, t)$ be spatiotemporal coordinates for trial k, M is length of trials, $S_k = \{k_1, \ldots, k_N\}$ be training set of trials, where $k_1, \ldots, k_N$ are corresponding trials for comparison. In our experiment, $M = 1850$, because normalized length of all trials was 7400 ms with sampling each 4 ms ($7400/4 = 1850$) and $N = 3$, because we had one instance of repeating trial in each block and we left one out for each comparison. For example, if we had one repeating trial in 4 instances with ids $9, 23, 45, 77$, we have $S_{23} = \{9, 45, 77\}$ etc. We computed spatiotemporal Gaussian centered around $\vec{x_j}$:

$$G_j((x, y, t)) = \frac{1}{(2\pi)^{\frac{3}{2}}\sigma_x\sigma_y\sigma_t} e^{-(\frac{(x-x_j)^2}{2\sigma_x^2} + \frac{(y-y_j)^2}{2\sigma_y^2} + \frac{(t-t_j)^2}{2\sigma_t^2})}$$

	Mean	SD
Easy	0.96	0.20
Medium	0.89	0.32
Hard	0.84	0.37

Table 3.1: Means and standard deviation for accuracy grouped by difficulty.

for some input vector $\vec{x} = (x, y, t)$, where $\sigma_x, \sigma_y, \sigma_t$ are parameters of Gaussian. Those parameters were set to values $\sigma_x = \sigma_y = 1.2°$ and $\sigma_t = 26.25$ ms and approximately correspond to size of fovea and mean length of short fixation (Dorr et al., 2010). Some spatiotemporal coordinates in scanpaths could be missing because of blink removal, we treated those missing data as if those coordinates were very far, so corresponding value of Gaussian would be zero.
Spatiotemporal fixation map is sum of Gaussians centered around vectors from S_k:

$$F_k(\vec{x}) = \sum_{i \in S_k} G_i(\vec{x}),$$

This fixation map was normalized to have zero mean and unit standard deviation. We get NSS map N:

$$N_k(\vec{x}) = \frac{F_k(\vec{x}) - \overline{F_k(\vec{x})}}{s(F_k(\vec{x}))},$$

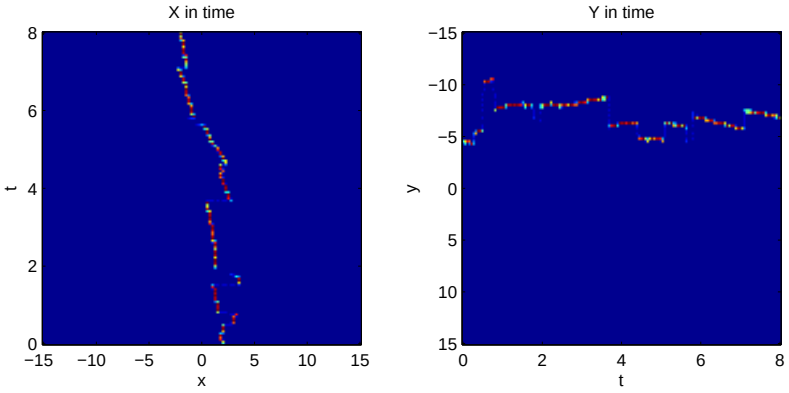
where $\overline{F(\vec{x})}$ and $s(F(\vec{x}))$ are sample mean and standard deviation. NSS value for trial k can be computed as value of NSS map divided

$$NSS_k = \frac{1}{M} \sum_{i=1}^{M} N_k^i(\vec{x_k}),$$

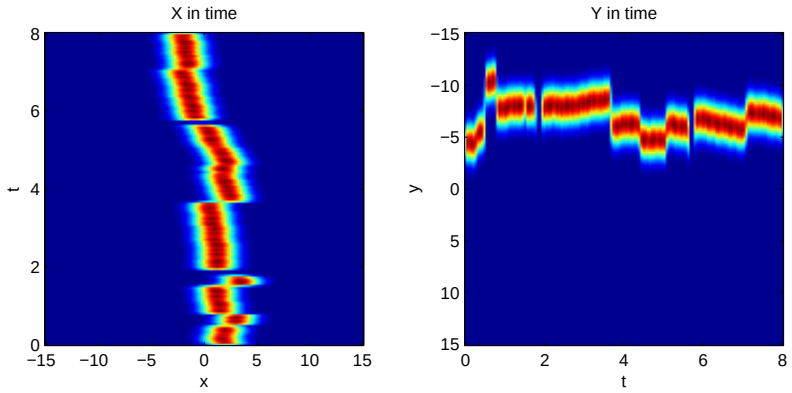
where $N_k^i(\vec{x_k})$ is i-th component of normalized fixation map for vector $\vec{x_k}$. We can see process of computing NSS values for trajectories from repeating trials in Figure 3.2

For better understanding of scaling of NSS values, we created a heatmap which visualizes how NSS decreases. In Figure 3.3 we compared two trajectories which differ only in combination of two parameters. First parameter represents overall distance of two trajectories where zero distance means that trajectories are the same (that means NSS value is high). Second parameter represents percentage of incoherency between two trajectories (percentage of scanpath, in which they are very distant – for those parts we get value of Gaussian approximately zero, while other parts are identical or varied by other condition). For example, point (0.5°,10%) corresponds to two trajectories which are distant 0.5° from each other in 90% of scanpath and they are very far from each other in 10%.

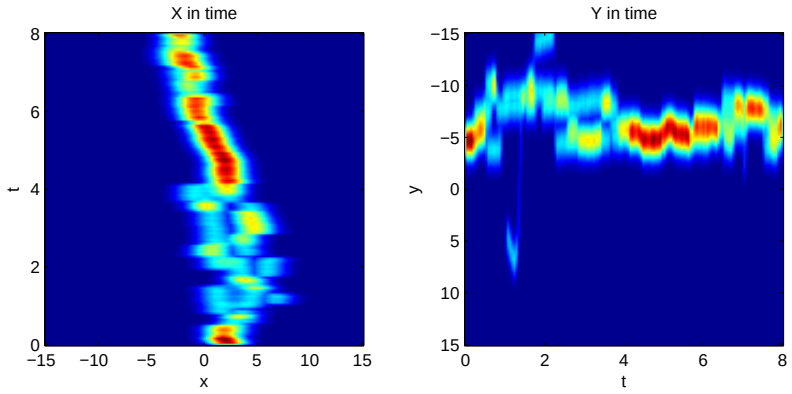
We implemented computation of NSS value as an operation on 3-dimensional matrices. Because algorithm works on discrete values, we reduced scanpaths to bins with size 0.25° for x and y coordinates and 50 ms for time coordinate. We could use a finer discrimination, but it would lead to similar results (Dorr et al., 2010).
We computed NSS for all repeating, changing and random trials. When computing values for changing trials, we used corresponding trials which differed in



(a) Simple scanpath



(b) Convolved trajectory with Gaussian



(c) Fixation map of several scanpaths.

Figure 3.2: Visualization of computing NSS value. Red areas represent parts of scanpaths which were fixated in several trials, light blue ones represent parts of scanpaths which were unique in one of trials.

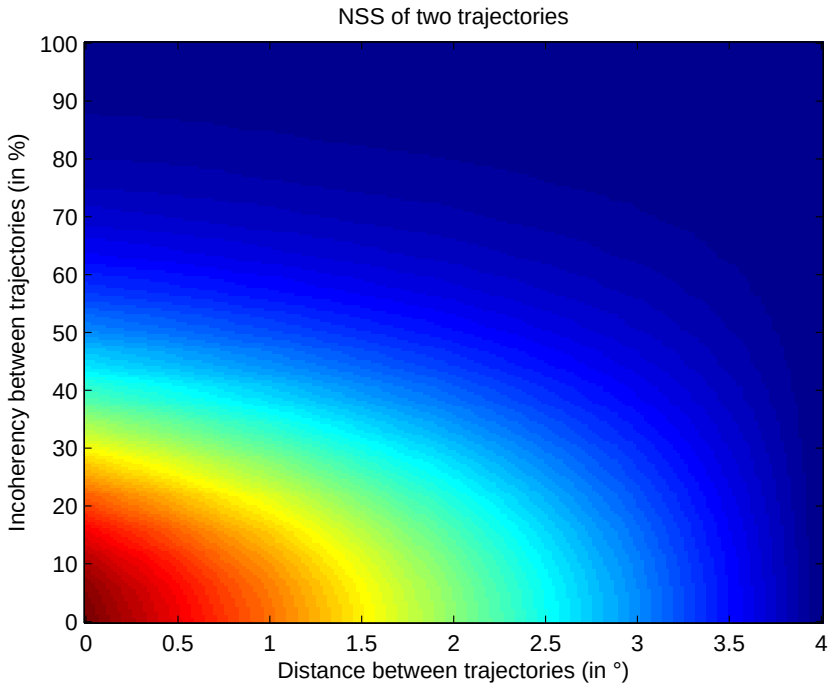


Figure 3.3: NSS values for two trajectories which varies in overall distance and in incoherence (percentage of trajectory in which they are very far from each other).

presented subsets. Random trials were only used as a baseline. To ensure comparable results when computing NSS for random trial, we did not use all other trials as S_k set, but we only randomly selected 3 from other available trials.
We used NSS values for measuring similarity of one scanpaths to the others (we could compute NSS with fixation map from only one another trajectory, but those values would have different scale); alternatively, we could talk about prediction of one trajectory from other and in this case NSS value would be a quantification, how good this prediction was.

Results of comparison

Statistical analysis for NSS values were done in program R (RDC Team, 2012). Basic descriptive statistic on Table 3.2 shows high standard deviation for repeating and changing trials.

	Mean	SD
Repeating	4.25	1.75
Changing	3.70	1.65
Random	0.40	0.59

Table 3.2: Mean values and standard deviations for each category

We compared NSS values of each trial type using one-way ANOVA. Test showed significant differences between types ($F(2, 87) = 198.4; p < 0.001$). Post-hoc tests have showed significant difference between repeating and changing trials ($t(510) = 3.378; p < 0.001$). We can see on Figure 3.4 significant decrease of

consistency between repeating and changing trials. This supports our hypothesis that if we increase number of distractors, eye trajectories will be affected. Detailed visualization of differences between repeating and changing trials can be seen on Figure 3.5. Post-hoc tests showed that difference between repeating and changing trials was significant for easy trials ($t(190) = 3.353; p < 0.001$) and for hard trials ($t(162) = 2.288; p = 0.023$).

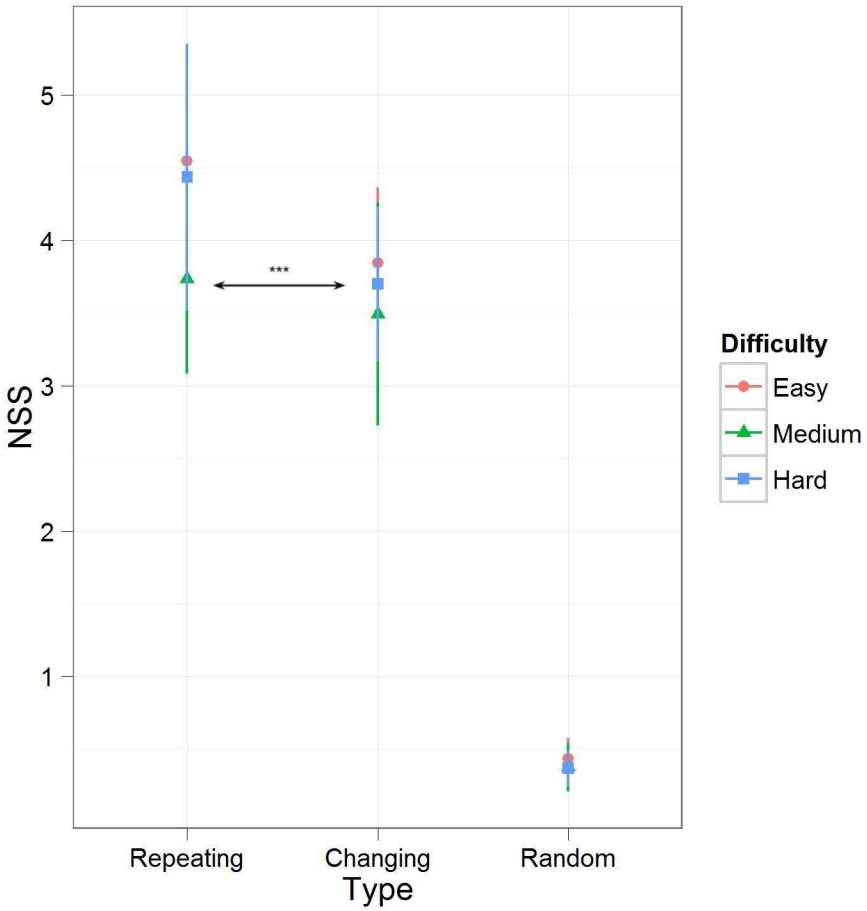


Figure 3.4: Difference among all types are significant. Consistency for random trials was used as a baseline value. We can see a significant decrease of consistency between repeating and changing trials, which supports our claim that the increase of distractors would affect eye movements. Vertical lines denote confidence intervals in each category.

We can see comparison of difficulties for repeating trials on Figure 3.6. ANOVA test showed no significant difference among difficulty categories for repeating trials ($F(2, 86) = 2.036$; $p = 0.137$). However, it seems there is a strange trend in consistency of eye trajectories. For easy and hard trials, consistency is slightly higher than for medium trials. We think consistency for easy and hard trials is high because in easy trials, there are not many other things to look at and in hard trials, the difficulty of tracking is so high that there are not other strategies to track targets. In medium trials there can be more strategies which could be used for tracking, so the consistency decreases. Our difficulty classification was, however, too gross, and we should use more participants for more solid data.

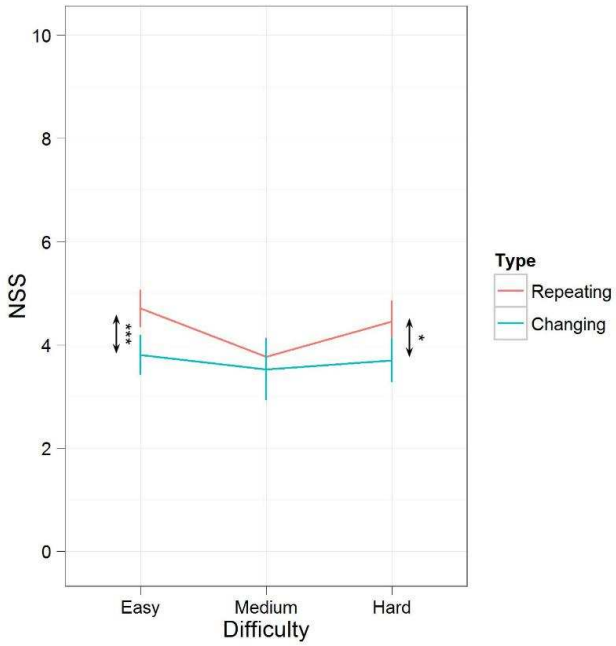


Figure 3.5: Detailed visualization of NSS values between repeating and changing trials. We left out random trials, they were only used as a baseline. The difference between easy and hard difficulties was significant. Vertical lines denote confidence intervals in each category.

3.4 Discussion

We designed an experiment which tried to determine if crowding had some effect on consistency of eye trajectories. It turned out that if a participant saw same trial repeatedly, consistency of his eye movements was high. For comparison we used trials in which number of distractors increased across blocks. If crowding did not affect tracking, additional distractors would not have effect on consistency of changing trials, so we should get similar NSS values for repeating and changing trials. NSS values for repeating trials and changing trials differ significantly so crowding somehow affects tracking. When we compared consistency of trials with different difficulty, we did not find out any significant difference, however it seems that there is decrease of consistency for trials with medium difficulty. We think that in general, multiple strategies can be used for tracking, but in trials with easy difficulty there are not enough distractors to use some sophisticated strategies and in trials with hard difficulty some tracking strategies could lead to bad tracking performance so participants do not use them. In trials with medium difficulty, several tracking strategies can be used while maintaining high tracking performance.

3.4.1 Repeating trials

Each repeating trial was presented once in each block. Because participants could notice that they had seen repeating trials, we wanted to randomize trial starts for each repeating trial, but we found out during data parsing, that we made error in programming, so each repeating trial was presented in each block in exactly

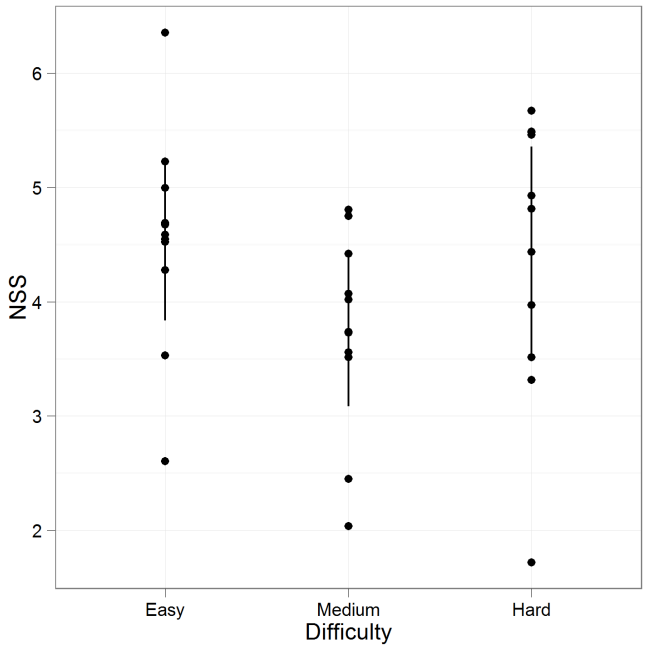


Figure 3.6: Difference between difficulty for repeating trials is not significant. We can see a decrease of consistency (although non-significant) for medium trials. Vertical lines denote confidence intervals in each category.

the same form. Three subjects (30% of participants) reported after experiment that they noticed some trials were repeating, but they did not pay attention to this fact.

3.4.2 Parsing data

We develop a heuristic for detecting blinks which used critical pupil size. This heuristic caught most of blinks in trials. We probably missed some blinks, because eye tracker is not accurate enough and pupil size of some blinks differ. We could not use higher critical value because in some trials pupil size differ significantly up to 70% of maximal pupil size. Some left out blinks would not affect NSS values a lot, because NSS is resistant to outliers. We discussed another method which computes numerical gradient and finds blinks by gradient value. This heuristic would not suffer so much from eye tracker inaccuracy. Another possibility would be to find blinks manually.

3.4.3 Statistics

We tested differences among trials using two one-way ANOVA (difference among trial types and difficulty for repeating trials). Another option was to use two-way ANOVA for testing, if interactions between trial type and difficulty were significant. In our experiment we had used relatively small group of participants. Because each participant could use different tracking strategy (some of them reported that they were trying to use specific strategy for tracking) validity of our findings could be affected.

3.4.4 NSS as measurement

In our experiment, we used normalized scanpath saliency as measurement of consistency one trajectory with others. With more scanpaths for creating fixation map, robustness of NSS to outliers increases. We only used 4 blocks in our experiment, so fixation map for each repeating trial was created from 3 scanpaths. It would be better to use more blocks next time.

3.5 Conclusion

We realized an experiment in which we studied influence of number of distractors on consistency of eye movements. It was shown that with increased number of distractors, consistency of eye movements decreased. We introduced NSS method for comparing eye movements and tried to describe meaning of its values. We will try to answer the question how exactly distractors influence eye movements in next chapter.

4. Models of eye movements

The question how scanpaths are consistent in dependence on number of distractors leads to the question how eye movements are related to the position of the objects. We wanted to find a formula which would predict eye movements. Our second goal was to try to train neural network to predict eye movements and compare this prediction with analytical model and behavioral data.

4.1 Related work on eye movements

There were several researches concerning eye movements strategies during MOT. Our approach to modeling strategies was based on work of (Zelinsky & Neider, 2008) and doctoral thesis of Fehd (2009). We also used some findings from Landry, Sheridan, and Yufik (2001)
Zelinsky and Neider (2008) studied eye movements during MOT in a realistic environment. They presented subjects a scene with 9 computer models of sharks moving in aquarium. Subjects tracked 1-4 targets and eye movements were recorded. Fixations were analyzed and each one was classified as either fixating to the target, fixating to the distractor or as fixation to the centroid of targets. They found out that tracking strategy people use depended on number of tracked targets. If people track one target they tend to fixate on that target, if they were tracking two or three targets they fixated the centroid of the targets. Interesting finding was that for four targets, people spent more time fixating targets then the centroid. Although people spent more time switching among targets then looking at the centroid, fixating centroid lead to better tracking performance. Authors hypothesized that we could find lost targets during tracking by comparing eye fixations with centroid positions of subset of targets. They also stated that people probably change strategies during tracking.
Fehd and Seiffert (2008) also studied strategies which could explain where participants looked. She classified strategies into three types:

- No motion – subjects did not move their eyes and stayed at roughly same position
- Tracking general motion – subjects pursued general motion of the targets
- Switching motion – subjects saccaded rapidly between targets.

They supported the original claim that people fixate centroid and discovered that people have better tracking accuracy when fixating on centroid of targets instead of switching between targets. By varying tracking speed they showed that preferring smooth pursuit over saccading is not a result of saccadic suppression (Fehd & Seiffert, 2010). Fehd also created centroid-target-centroid strategy which leads to better tracking accuracy when used. People following this strategy switch between centroid and targets.
Landry et al. (2001) studied eye movements in airplane tracking. Participants have to selected planes entering and leaving tracked area and have to spot collisions. They found out that subjects spent more time looking on the planes which

were going to collide then on planes flying safely.
The findings so far support hypothesis that people create from targets virtual object (Yantis, 1992) and they fixate its center. We think that in simple configurations of MOT task, fixating on centroid could be good strategy. However in configuration with more distractors, this strategy could be suboptimal.

4.2 Analytical models

As we discovered in the experiment, scanpaths are influenced by number of distractors. All the presented strategies work only with targets positions so they would not explain this change of scanpath. We defined several natural conditions that should valid tracking strategy entail:

- If one target is presented (no distractors), eyes should fixate on it
- If two targets are presented (no distractors), eyes should fixate somewhere between them
- If three or more targets are presented (no distractors), eyes should fixate into the convex hull of targets
- If we have target-target-distractor in one line, eyes should fixate nearer the target with distractor nearby

We want to discuss only strategies which are supported by behavioral data. Centroid looking strategy fulfills those criteria.
If we express *centroid* strategy, we will get

$$\vec{x} = \underset{\vec{x'}}{\operatorname{argmin}} \sum_{t \in T} \|\vec{x'} - \vec{t}\|,$$

where T is a set of targets. We can compute centroid for all dots, not only targets, we will refer to this strategy as *all-centroid strategy*.
In visual perception, when target and distractor gets close to each other in periphery, we can mix them up due to crowding phenomenon. We created several strategies that try to minimize effect of crowding during tracking. We will denote them as crowding minimizing strategies (or simply *crowding strategies*) which tries to minimize distance. We created two variants of minimizing crowding distance:

$$\vec{x} = \underset{\vec{x'}}{\operatorname{argmin}} \sum_{t \in T} \sum_{d \in D} \frac{\|\vec{x'} - \vec{t}\|}{\|\vec{t} - \vec{d}\|} \tag{4.1}$$

$$\vec{x} = \underset{\vec{x'}}{\operatorname{argmin}} \sum_{t \in T} \sum_{d \in D} \left(\frac{\|\vec{x'} - \vec{t}\|}{\|\vec{t} - \vec{d}\|} \right)^2, \tag{4.2}$$

where T is set of targets and D is set of distractors. We used quadratic variant (4.2), because it could produce more plausible predictions. If we had only two targets on coordinates (x, y) and $(x + n, y)$ and one distractor on coordinate $(x + \frac{n}{2}, y + m)$, where m, n are arbitrary values, linear variant would predict any of points between (x, y) and $(x + n, y)$ as equally plausible while quadratic would

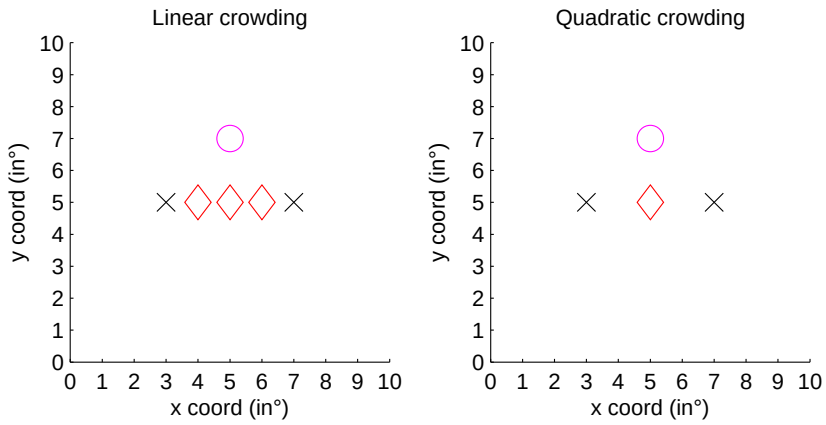


Figure 4.1: Difference between linear crowding and quadratic crowding strategy. Crosses represent targets, circle represents distractor, and diamond represents predicted eye locations. Linear crowding evaluates all three locations as equally possible, while quadratic prefers central point.

predict point $(x + \frac{n}{2}, y)$. We can see difference between those two strategies on Figure 4.1. As there were more objects in the real trials, there would be usually only one minimum. We could use other powers but the differences would be smaller then eye tracker measurement error. We tried linear and quadratic variants which maximized sum of $\frac{\|\vec{t}-\vec{d}\|}{\|\vec{x}-\vec{t}\|}$ but those strategies strongly preferred targets over anything else so we did not test them further. Because people are not able to keep identity of objects during tracking, we only measured distance between targets and distractors and not between targets mutually. If two targets occlude, it does not matter which one is which, so it is not important to minimize this distance for successful tracking.

Because crowding occurs more on periphery then on the fovea, we tried variants of crowding strategies which did not use all distractors D, but only those which were closer to the current eye position then Bouma distance ($d \in D \wedge \frac{\|\vec{x}-\vec{t}\|}{\|\vec{t}-\vec{d}\|} \leq \frac{1}{2}$). We had four variants of crowding strategies in total: linear, quadratic, linear with Bouma's distance cutoff, quadratic with Bouma's distance cutoff.

We compared each of four crowding strategies with centroid strategy, all-centroid strategy and *constant* strategy which predict all eye fixations into the center of screen. We used NSS metric for comparison, because we wanted to get similar values as in our experiment. NSS values were computed for all trajectories used in repeating trials and we compared the predicted trajectory with fixation maps consisting from eye trajectories from corresponding repeating trials. We can see visualized predicted trajectories on Figure 4.2.

As Zelinsky and Neider (2008) pointed out, tracking strategy that people use can be dependent on the type of task. In our experiment we changed number of distractors, so we wanted to find out if consistency trajectories predicted by strategies was dependent not only on the type of strategy, but on difficulty settings as well. We computed NSS values for each strategy on repeating trials and we compared strategies using two-way ANOVA (factors: strategy type and diffi-

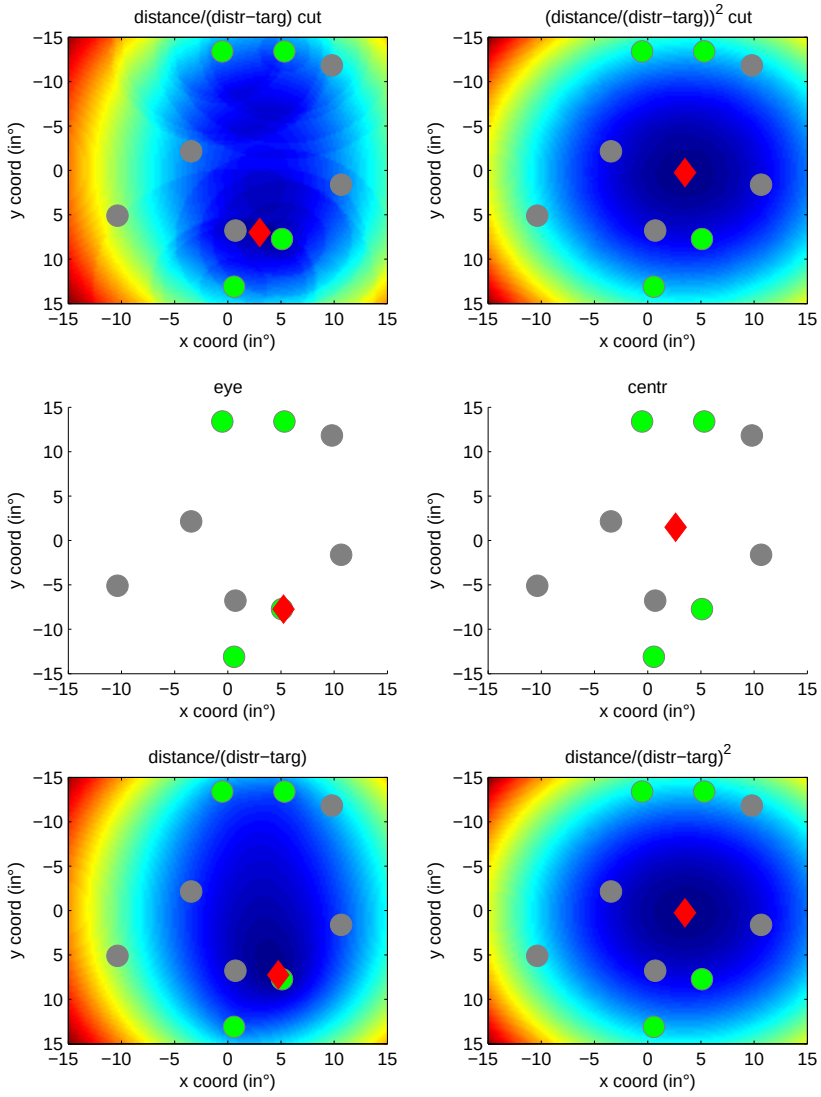


Figure 4.2: Predicted eye positions for each strategy. Targets are green dots, distractors are gray, red diamond represents predicted eye position. Heatmap represents values of functions for crowding strategy (blue areas have minimal value). In the middle plots we can see eye positions from experiment and centroid strategy for comparsion. In this specific frame we can see that linear variant is very similar to eye positions from experiment.

culty category). We used 7 types of strategies: four crowding strategies, centroid, all-centroid and constant strategy. It turned out that NSS value is strongly dependent on used strategy ($F(6, 189) = 249.86, p < 0.001$) and independent on number of distractors ($F(6, 189) = 1.68, p = 0.189$). This means that prediction strength of strategies differs and those strategies predict equally well independently on number of distractors. There is no interaction between strategy and number of distractors ($F(6, 189) = 16.45, p = 0.156$). This means that there is not strategy that predicts significantly better for some difficulty category. We

can see all results in Table 4.1. On Figure 4.3 we can see comparison of NSS

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
StrategyType	6	249.86	41.64	43.20	0.0000
Difficulty	2	3.25	1.62	1.68	0.1885
StrategyType:Difficulty	12	16.45	1.37	1.42	0.1588
Residuals	189	182.18	0.96		

Table 4.1: Results of two-way ANOVA. We can see that consistency of predicted trajectories is strongly dependent on used strategy.

values dependent on strategy and difficulty. Linear crowding strategy seems to predict eye movements for all difficulty settings better then other strategies. Cut off variants of crowding strategies predict eye movements better in trials with more distractors (this difference was significant, e.g. for quadratic crowding difference between easy and hard trial was significant – $t(15.40) = -2.54; p = 0.02$). We think that this could be explained by limited capabilities of visual system. With increasing number of distractors, its computably demanding to process all distractors, so only distractors in proximity of targets are processed.

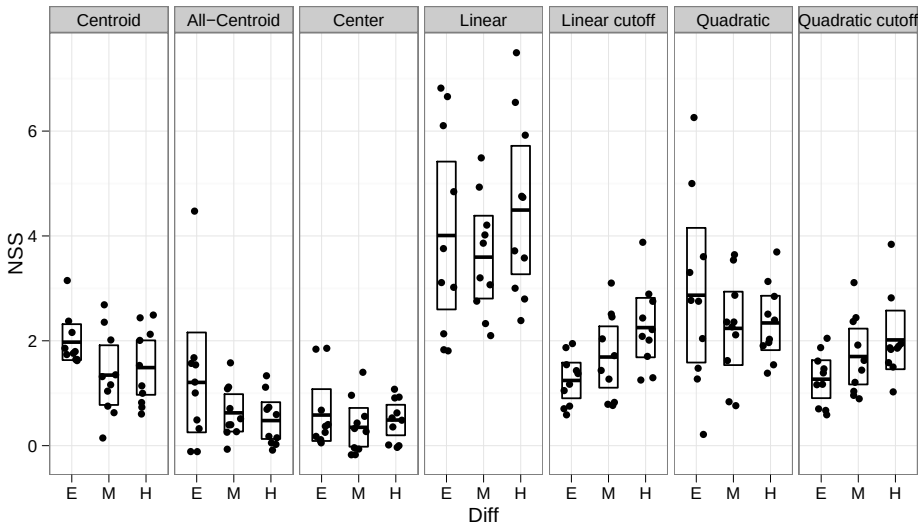


Figure 4.3: Plot of NSS values dependent on strategy and difficulty settings (E for easy, M medium and H for Hard). Linear crowding strategy have largest NSS values.

We tested, how well predict linear crowding strategy eye movements in comparison with real eye trajectories. As we can see on Figure 4.4, consistency of predicted trajectories by linear crowding strategy was similar as real eye data, difference between them was not shown as statistically significant ($F(2, 54) = 0.33, p = 0.566$). It means that we can take trajectory predicted by linear crowding strategy and it will predict eye movements similarly well as real eye trajectory. Whole results of ANOVA can be seen on Table 4.2.

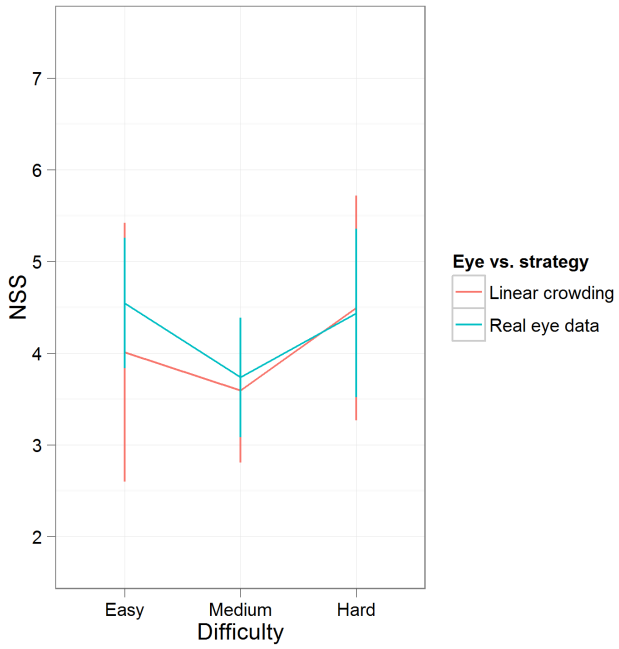


Figure 4.4: Differences between real eye data and linear crowding strategy. Differences between eye and strategy are not significant. Vertical lines denote confidence intervals in each category.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
StrategyOrEye	1	0.64	0.64	0.33	0.5655
Difficulty	2	6.99	3.50	1.82	0.1711
StrategyOrEye:Difficulty	2	0.92	0.46	0.24	0.7875
Residuals	54	103.46	1.92		

Table 4.2: Results of two-way ANOVA. Differences in consistency of predicted trajectories between another eye data from repeating trial and between linear crowding strategy were not significant.

4.2.1 Discussion

Linear crowding strategy has NSS values comparable to real scanpaths. It means that if we want to predict where a participant will look on the frame, we could use either eye positions from another repeating trial of the same configuration, or we can use scanpath predicted by linear crowding strategy. We do not claim that linear crowding strategy is biologically plausible, minimization of Equation 4.1 uses all distractors, but as we found out, prediction of cut off versions of crowding strategies increases with number of distractors. It would be a logical step to modify experiment and see, if there is a threshold in number of distractors, where crowding strategies without cut off would be worse then cut off versions. Tracking accuracy was good even for many distractors (see Table 3.1), so we could still use MOT to prove this claim.
Our dot trajectories used in MOT were generated randomly. If we compute centroid of targets, it will remain near the center of the rectangle, in which targets

moved. Differences between strategies will be relatively small, so it will be good idea to try to create dot trajectories, which differ more significantly.
In our crowding strategy, we only use dot positions in current frame. We did not encompass previous positions because as stated before, people probably did not predict dot positions during tracking (Alvarez et al., 2005) and we wanted to created strategies similar to centroid strategy which uses information with current frame only as well.
Linear crowding strategy mainly explains variability of eye movements using smooth pursuit. Switching between several places is modelled only, if two local minima are in two nonadjacent places and in each frame one of them became minimal. This situation however occurs rarely so in general linear crowding strategy can not predict eye movements which switch between several places. We will discuss another possible strategies in general discussion.

4.3 Neural network models

Linear crowding strategy can be used for predicting eye movements very well. We wanted to try out, if we can train neural networks using behavioral data to predict eye movements. It should be possible, because if can tracking strategy be expressed as function of dot positions, neural network should be able to approximate this function. In real scenarios, human eye movements do not follow some simple strategy, but they contain many unpredictable artifacts e.g. participant may get tired and look to some random part of the screen. One of the common hardly predictable patterns in eye movements is *rescue saccade*. Zelinsky and Todor (2010) found out that people during tracking tend to make rescue saccades - saccades to targets which are in danger of being lost because of occlusion. We will consider such artifacts in movements as noise. We expect that learning such patterns would be hard, so we had to develop some methods for preprocessing the data to reduce the noise. For creating and training neural networks, we used Neural Network Toolbox for MATLAB (Demuth & Beale, 1992).

4.3.1 Description of Neural Network Toolbox

Neural Network Toolbox is a collection of functions and objects for using neural networks. It has functions for every step of using neural networks - from preparing data to predicting output for unknown dataset. We can use it to create many different network types like simple perceptron, multi-layer perceptron, RBF networks, LVQ networks or even some recurrent networks like Hopfield's network. We can select many parameters of neural network like number of hidden neurons and layers, activation functions between layers, performance functions for evaluating error of networks and most importantly we can select from many different learning algorithms. It includes functions for data normalization which are automatically applied to the new set of data if selected. Toolbox divides data set into train/validation/test and we can set this ratio. Toolbox uses graphical output and/or output to the console, so it can be easily used in some script or in the function.
For our purposes we used MLP network. Neural Network toolbox has efficient

built-in algorithms for training MLP network. Learning can be done either *batch training* - all weights are adjusted after presenting all training data or *incremental training* - networks adapts its weights after each sample. In general batch learning is usually more efficient but requires more memory. Both batch and incremental training have parameters which control conditions when training should stop. Most important criteria to stop learning are:

- maximum number of cycles of training algorithm
- minimal performance error between network outputs and desired outputs from dataset
- minimal gradient of learning function during training
- maximum number of succeeding error increase in validation set
- maximal time for which a network can learn

There are many built-in learning algorithms available. It is recommended to use Levenberg-Marquardt (L-M) algorithm for small and medium sized networks and scaled conjugate gradient for large networks. It is possible to use more then one hidden layer, but its approximation capabilities are nearly same as MLP network with one hidden layer.

4.3.2 Description of used MLP network

In all our experiments we used neural network with one hidden layer with 30-50 neurons. We wanted to train MLP network to predict eye positions for current frame, so our output layer has 2 neurons (x and y coordinates) and input layer has $2n$ neurons where n is number of dots (each dot has two coordinates). First 8 neurons were positions of the targets $(x_1, y_1, \ldots, x_4, y_4)$. Neural networks can work only with inputs of fixed length so we could not use all eye trials as training data, because we had different number of distractors in each trial. We decided to use only trials with configuration 4:4, because this configuration is commonly used in MOT experiments. We used random and changing trials for training the MLP network and repeating trials were used for testing and computing NSS values. Even with this constraint on number of distractors, we had very large data set. We had 46 trials with configuration 4:4, each trial had 1850 samples, so we had data set with more than 80000 samples. Because eye tracker recorded approximately three samples for each video frame, we used only $\frac{1}{3}$ of eye data, because we did not want to have duplicates in out training set.
We used L-M algorithm for batch training, tangent sigmoid as activation function and mean square error without regularization as performance function. Tangent sigmoid has similar properties as logistic sigmoid and it was preset in toolbox. Before learning, data was mapped to interval (-1,1) and randomly divided into train, validation and test sets in ration $0.70 : 0.15 : 0.15$.

4.3.3 Learning artificial data

Before we tried to learn MLP network to predict eye movements, we decided to create some artificial data and see, if neural network were capable of learning

analytical strategies presented in previous section. We created three artificial datasets, on which we trained neural network. They were based on strategies which could explain eye movements during tracking:

- centroid strategy dataset – outputs of centroid strategy dataset was created from data by averaging values of x and y coordinates of targets.
- switching strategy dataset – outputs of switching strategy dataset was created from data by dividing targets into two subsets and, computing centroids of those two subsets and then each switch between them each 20 frames
- linear crowding dataset – outputs of linear crowding dataset was computed using linear crowding strategy

All datasets used same randomly generated matrix with inputs (50000 samples). We expected that MLP network should learn centroid strategy perfectly, but it should not be able to learn switching between targets. Switching between targets is a strategy which incorporates time information. If we had two exact succeeding frames and in each frame eyes were fixating on different centroid, we could not predict, where subject would look without time information, how long he has fixated on one target. Crowding strategy was expected hard to learn as well, because MLP network can approximate only continuous functions. We computed crowding strategy numerically for each frame, so it appears to network as noncontinuous function (if there are two local minima in distant areas of visual fields then even small difference in dot position can lead to switching gaze between those minima). Another problem with learning linear crowding strategy is its unboundedness: We can express linear crowding strategy for n dots as function of $n+2$ parameters $x, y, \vec{t}_1, \ldots, \vec{t}_k, \vec{d}_1, \ldots, \vec{d}_{n-k}$, where k is number of targets (4 in our case), $\vec{t}_i$ are coordinates of i-th target, $\vec{d}_i$ are coordinates of i-th distractor. Function is not defined for cases where $\|\vec{t}_i - \vec{d}_j\| = 0$ for some i, j and its value approaches infinity for inputs where targets and distractors are close ($\lim_{\vec{t}_i \to \vec{d}_j} \frac{\|\vec{x}-\vec{t}_i\|}{\|\vec{t}_i-\vec{d}_j\|} = \infty$). Results show that MLP network (30 neurons in hidden layer, L-M training algorithm, without regularization) was able to learn centroid strategy very accurate (mean error on test dataset was $< 10^{-7}$), but it was not able to learn switching at all (mean error on test dataset was ~10) and it learned crowding strategy poorly (mean error on test dataset was ~1). Differences between crowding strategy and predicted outputs sometimes differed more then 6°. We tried MLP network with more neurons in hidden layer and used other training algorithms, but it did not improve the performance. Often used technique for increasing learning capabilities of neural networks is to add polynomial combinations of inputs (e.g. we add inputs x_1x_2, x_1^2, x_2^2 for inputs x_1 and x_2), but it increased size of input layer quadratically (we had 16 inputs, so this operation adds 16^2 more) and we were not able to train this network even with fast training algorithms.

Training MLP network on artificial data has shown that neural networks should learn some pattern in eye movements (centroid) but it could have problems with more complex strategies. In order to successfully train neural network to predict eye movements, we had to use some techniques to reduce noise in the data.

4.3.4 Smoothing the data

Typical eye data during MOT usually can not be explained only by simple tracking strategy. People typically do a lot of unpredictable movements and we have to simplify data in order to successfully train neural network. Our first approach was smoothing the data. Smoothing is process of approximating data with some function. We tried to use following algorithms for smoothing (their main difference is selection of approximation function) :

- Moving average – Moving average is simple method for smoothing data. Each value of x_i and y_i coordinates is substituted by mean of $k/2$ preceding and $k/2$ succeeding values.
- Savitzky–Golay smoothing filter – Savitzky and Golay (1964) created method similar to moving average, but coefficients for neighbours' values are computed from unweighted linear least-squares regression and a polynomial model of specified degree. Main advantage against moving average is that SG filter preserves features of the distribution (minima, maxima, etc.).
- Local regression (loess) – Cleveland (1979) have proposed method for smoothing data using local regression which weights neighbours' values with a 2nd degree polynomial model. Robust version of loess assigns lower weights to the outliers.

There are other smoothing functions, but our goal was not to find an ideal smoothing function, but only to find one, which reduces unwanted eye movements. We compared smoothed data from each algorithm and decided to use robust version of local regression (rloess), because it smoothened most unwanted eye movements like rescue saccades. Typical result of smoothing functions can be seen on Figure 4.5.

Smoothing was able to remove most of the eye movements, which could make learning more difficult, but it had problem with those eye trajectories which switch between several places (typically between two places). We could increase degree of polynom for rloess to create straight line from switching eye movements, but it would lead to the loss of information in non-switching eye data. We decided to find and discard trajectories with too much switching. Saccade between two places leads to a big change in position in small time. We computed numerical gradient in each dimension for comparing change of direction. Then we get value $g = \sum |\vec{g}_x| + \sum |\vec{g}_y|$ for each trajectory, where $\vec{g}_x$ and $\vec{g}_y$ are numerical gradients for x coordinates and y coordinates. We visualized data and set threshold value $g = 150$ for trajectories with too much switching. We can see typical trajectory with too much switching and gradient values on Figure 4.6. This data reduction discarded 25% of the trajectories, but it still left enough data for training.

4.3.5 Increasing variability and rounding outputs

The variability of data set for MLP network was very low. Dot positions and eye positions in each two succeeding frames differed only a little. In order to train the network successfully we had to increase the variability of data. New data could be created simply by permuting targets and distractors. Network should predict

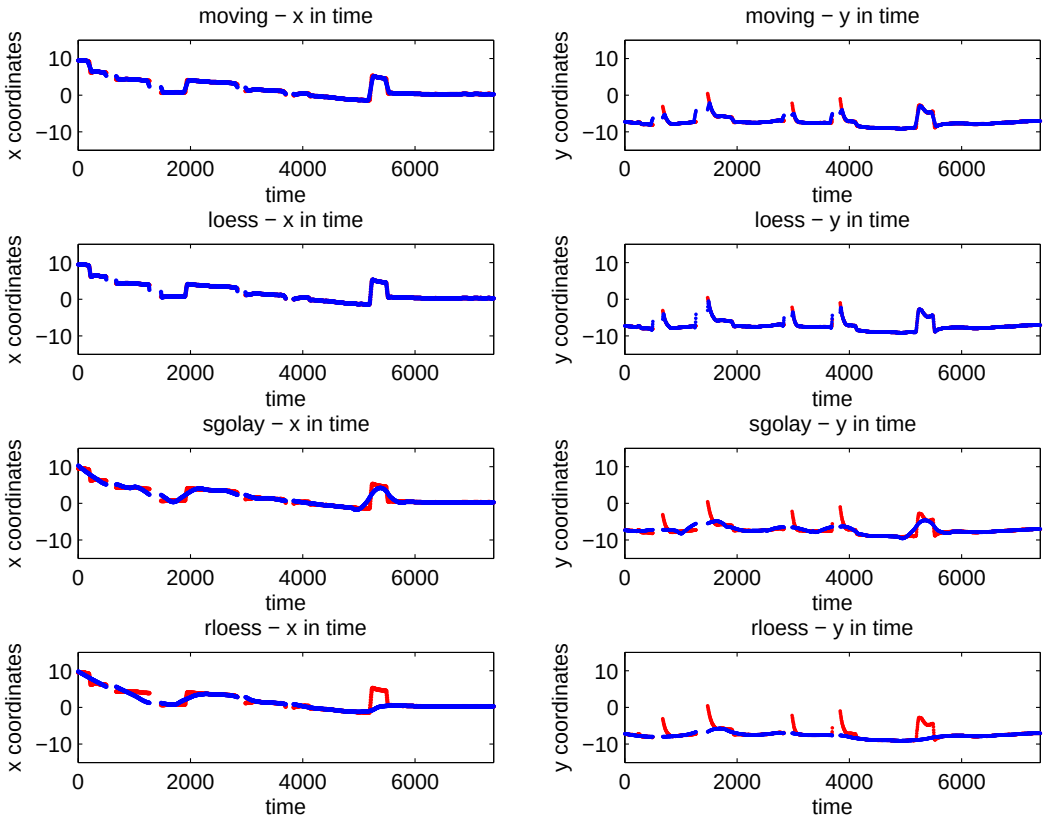


Figure 4.5: Results of smoothing the data set with several algorithms. Red line represents original data and blue line represents data after smoothing. For our purposes is most suitable robust version of local regression (rloess).

eye movements correctly if we have inputs in order $x_1, y_1, x_2, y_2, x_3, y_3 \ldots, x_8, y_8$ or $x_2, y_2, x_1, y_1, x_3, y_3, \ldots, x_8, y_8$. Without this operation, network would learn order of the inputs which could lead to the poor performance. We had to permute inputs separately among targets and among distractors, if we changed order of x_4, y_4 and x_5, y_5, we would get different scene, because network would predict eye position based on three targets and one distractor. Permuting input lead to increase of data ($4!4!n$, where n is number of samples in original data set), so we used only portion of data of each permutation.
As stated before, eyes have natural dispersion so two fixations close to each other could differ only because of inaccuracy if in motor system of an eye. We decided to round outputs down to the precision step of 0.25°. This classification was quite rough, but we wanted to compute NSS value, so we would round those values to the bins anyway.

4.3.6 Enlarging dataset for testing

Because we could use only trials with 4:4 configuration, we got just a small dataset, which we could use for testing. In order to get more robust results, we decided to create new data using symmetry of the visual field. We could get 3 new trajectories from each trajectory in dataset using axial symmetry (horizontal, vertical and combination of both). For example, if we had trajectory with dot positions $\vec{x}_1, \vec{y}_1 \ldots, \vec{x}_n, \vec{y}_n$ and eye positions eye_x,eye_y, we could get a new sample

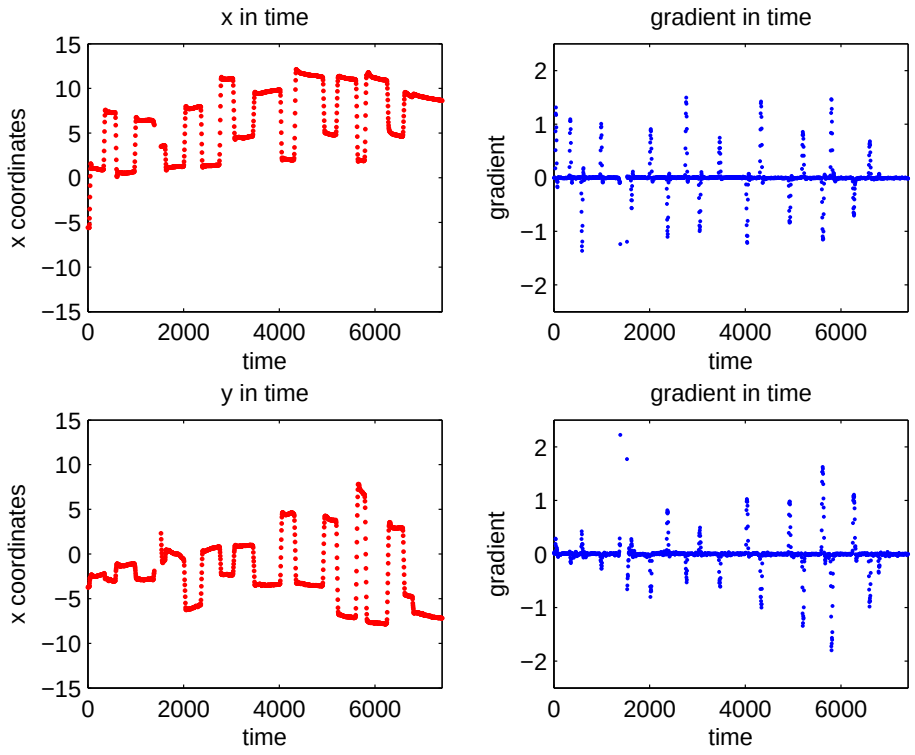


Figure 4.6: Example of eye movements with a lot switching between several places. If we smooth this function, it would differ too much from original eye data. In left column we can see trajectories in time, in the right plot are computed values of the gradient of trajectory.

$\vec{x}_1, -\vec{y}_1 \ldots, \vec{x}_n, -\vec{y}_n, eye_x, -eye_y$ using horizontal symmetry.

4.3.7 CrowdMOT02 experiment

In order to get a unique dataset for testing neural network, we created a new MOT tracking experiment with similar parameters as CrowdMOT. We will refer to it as CrowdMOT02. This experiment was only used for collecting validation data, so no hypotheses were formulated.

Differences between CrowdMOT and CrowdMOT02

In CrowdMOT02 experiment 40 trials divided into 4 blocks were presented to participants. We only needed small dataset, so we had three participants only. There were two types of trial: repeating and random. They were same as in CrowdMOT experiment (we did not use changing trials). In each block there were 6 repeating trials and 4 random trials. All trials have 4:4 configuration, so there were no difficulty categories. All other parameters were same as in CrowdMOT experiment.

Data parsing and analytical strategies

We have to exclude 4% of data because of eye tracker error. As a criterion for blink removal, we used threshold pupil size. Overall tracking accuracy was 90% (lowest accuracy was 87.5%) so we did not have to exclude any participant.
We computed analytical strategies and as we can see on Figure 4.7, NSS values for strategies are similar as NSS values for easy task in in CrowdMOT experiment.

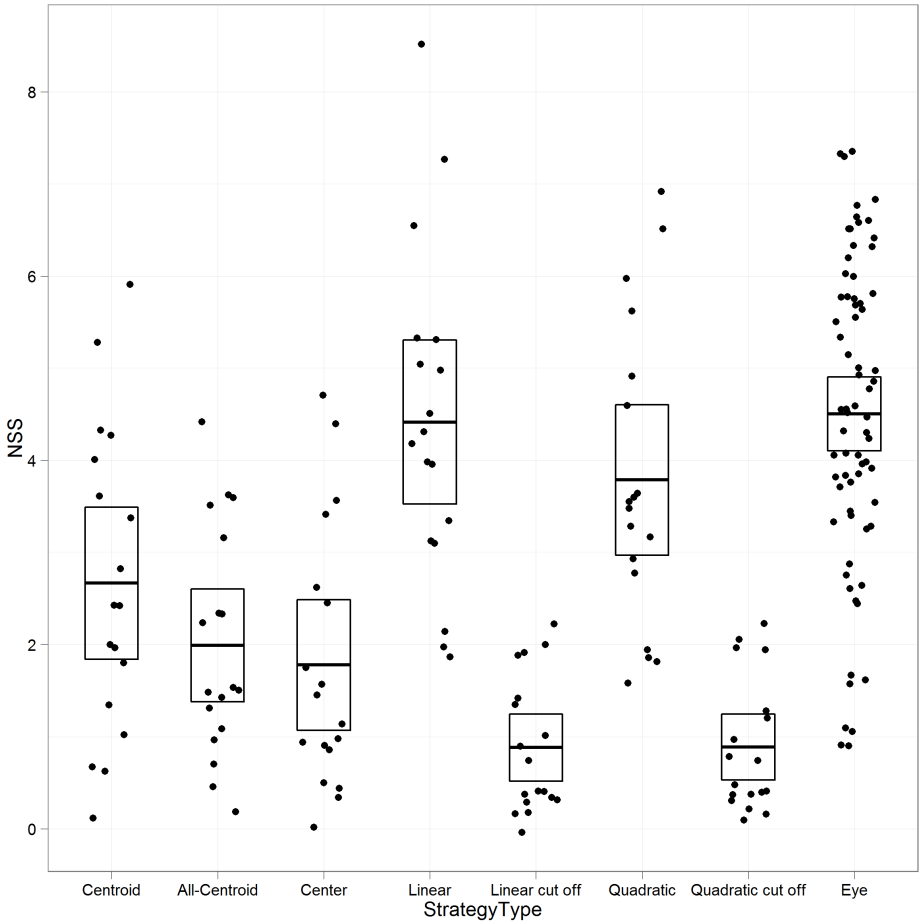


Figure 4.7: Plot of NSS values for dependent on strategy and difficulty settings. NSS values look similar as in CrowdMOT experiment. We added NSS values of repeating trials for comparison. We can see that values are similar to values for linear strategy.

4.3.8 Results

Training the networks

We trained neural network (with structure described in previous part) for each combination of presented data adjusting. Following combinations were tried:

- Use of raw data
- Smoothing data using rloess

- Removal of the data with too much switching (using gradient threshold as stated before)

For each of three variant we varied following operations:

- No operation
- Permutation of the data
- Rounding outputs
- Permutation of the data and rounding outputs

Last parameter we varied was the size of the hidden layer, we used 30 and 50 neurons so we trained $3 \times 4 \times 2$ MLP networks in total. For each trained network NSS values were computed. Fixation map for computing NSS was created from repeating trials same as for analytical models. Training of networks took several days because of limited access to computers with Neural toolbox installed, so we were able to train each network only once. We are aware that those results could be affected by random initialization of weights, so it would be correct to run those calculations several times to get more robust results.

Testing networks

We tested values on two datasets: enlarged dataset from original CrowdMOT experiment and dataset from CrowdMOT02 experiment. Because same method was used for trajectory generation in both cases, we expected the same results. We can see results for CrowdMOT dataset on Table 4.3 and results for second experiment on Table 4.4. We could not compute analysis of variance on those results, because we did not have enough samples, however we are able to see differences between prediction strength of trained MLP network. We will use notation $mean(Subset) = val$, which represents mean value of rows in table from specified subset. For dataset from CrowdMot experiment, we got best results for networks which used permuted inputs for training and had rounded outputs ($mean(PermutedAndRounded) = 1.282$). Smoothing helped a little ($mean(SmoothAndGrad \vee SmoothOnly) = 1.185$, $mean(Raw) = 1.004$), whereas the increase of number of neurons in hidden layer did not improve prediction, it even led to worse values of NSS ($mean(30) = 1.190$, $mean(50) = 1.059$). Analysis of second dataset showed similar results - best operation was permutation of samples and rounding outputs ($mean(PermutedAndRounded) = 1.324$), more neurons in hidden size did not help ($mean(30) = 0.912$, $mean(50) = 0.474$). In this dataset, smoothing had no effect on training the network ($mean(SmoothAndGrad \vee SmoothOnly) = 0.670$, $mean(Raw) = 0.739$).
It seems that if we take permuted and smoothed samples with rounded outputs as a training set, we will get best results ($mean(PermutedAndRounded \vee (\neg Raw)) = 1.469$, mean was taken from both tables). Mean NSS value for predicted trajectories by trained MLP network was worse than NSS value of trajectories predicted by centroid strategy (as we can see on Figure 4.3). It means that trajectories created by centroid strategy were more consistent with eye data than those trajectories predicted by trained neural network.

	TypeInput	TypeOperation	HiddenSize	NSS
1	SmoothAndGrad	NoOperation	30	0.89
2	SmoothAndGrad	Rounded	30	**1.71**
3	SmoothAndGrad	Permuted	30	1.15
4	SmoothAndGrad	PermutedAndRounded	30	1.36
5	SmoothAndGrad	NoOperation	50	0.96
6	SmoothAndGrad	Rounded	50	0.89
7	SmoothAndGrad	Permuted	50	1.07
8	SmoothAndGrad	PermutedAndRounded	50	1.35
9	SmoothOnly	NoOperation	30	0.96
10	SmoothOnly	Rounded	30	1.10
11	SmoothOnly	Permuted	30	1.05
12	SmoothOnly	PermutedAndRounded	30	**1.58**
13	SmoothOnly	NoOperation	50	0.95
14	SmoothOnly	Rounded	50	1.27
15	SmoothOnly	Permuted	50	1.28
16	SmoothOnly	PermutedAndRounded	50	**1.38**
17	Raw	NoOperation	30	0.83
18	Raw	Rounded	30	1.38
19	Raw	Permuted	30	1.24
20	Raw	PermutedAndRounded	30	1.03
21	Raw	NoOperation	50	0.82
22	Raw	Rounded	50	0.84
23	Raw	Permuted	50	0.90
24	Raw	PermutedAndRounded	50	1.00

Table 4.3: Summary of mean NSS values for each trained network using enlarged data from CrowdMOT experiment. Permuting and rounding data with combination of smoothing data had best influence on scanpaths prediction. More neurons in hidden layer did not have an influence on prediction. Three best values are displayed in bold.

4.3.9 Discussion

We were able to train neural network to predict eye movements for MOT task. If we computed NSS values (with fixation map created from repeating trials) for predicted trajectories, we get smaller values than we got for centroid strategy. We found two possible explanations of this phenomenon.

First, it could mean that we did not remove all of the noise in data, so network could not learn tracking strategy properly. If there were some artifacts unrelated to dot position in current frame, it would provide misleading training data and network would stop its training because of validation checks. This claim is supported by fact that best operation that increased prediction capabilities of the network was permuting the data which generated more training samples and helped to better discriminate noise in the data. It would be possible to use a technique with visual field symmetry as another operation which we could use for increasing variability in data, but we wanted to have some method for enlarging dataset for testing. If we had more data, it would be a good idea to use this

	TypeInput	TypeOperation	HiddenSize	NSS
1	SmoothAndGrad	NoOperation	30	0.12
2	SmoothAndGrad	Rounded	30	0.81
3	SmoothAndGrad	Permuted	30	0.56
4	SmoothAndGrad	PermutedAndRounded	30	1.33
5	SmoothAndGrad	NoOperation	50	-0.14
6	SmoothAndGrad	Rounded	50	-0.18
7	SmoothAndGrad	Permuted	50	0.92
8	SmoothAndGrad	PermutedAndRounded	50	1.13
9	SmoothOnly	NoOperation	30	-0.12
10	SmoothOnly	Rounded	30	0.84
11	SmoothOnly	Permuted	30	**1.73**
12	SmoothOnly	PermutedAndRounded	30	1.61
13	SmoothOnly	NoOperation	50	-0.20
14	SmoothOnly	Rounded	50	0.50
15	SmoothOnly	Permuted	50	1.11
16	SmoothOnly	PermutedAndRounded	50	0.71
17	Raw	NoOperation	30	-0.13
18	Raw	Rounded	30	0.70
19	Raw	Permuted	30	**1.87**
20	Raw	PermutedAndRounded	30	**1.63**
21	Raw	NoOperation	50	-0.20
22	Raw	Rounded	50	-0.20
23	Raw	Permuted	50	0.71
24	Raw	PermutedAndRounded	50	1.54

Table 4.4: Summary of mean NSS values for each trained network using data from CrowdMOT02 experiment. Permuting and rounding data had best influence on scanpaths prediction. Smoothing did not have influence on prediction. More neurons in hidden layer did not have an influence on prediction. Three best values are displayed in bold.

method for increasing variability.
Second, tracking strategy could be too complex for MLP network to learn. We were able to train network properly for centroid strategy but we were unsuccessful for crowding strategy. MLP networks should be able to learn continuous functions with real valued bounded range, but we showed that linear crowding strategy has unbounded range. This could mean, that more sophisticated strategies which explain tracking better than centroids can not be learned by neural networks.
One of possibilities for improvement would be to use time delay neural networks which are often used for time series prediction. They have same structure as MLP networks, but instead of one sample we present last k samples as input. Although this could help with learning some sophisticated tracking strategies, we did not want to use time information for learning. Predicting eye position based only on positions of objects helps us determine if tracking could be explained only by positions of dots in the frame. Predicted trajectories from MLP network are

comparable with analytical strategies, because they work with same information, time delay neural networks have extra information about last positions of dots. Performance of neural networks was worse than analytical model for predicting trajectories, but because of high explaining capabilities of analytical models, it would be surprising if neural networks predicted trajectories even better.

4.4 General discussion and conclusion

We have studied eye movements during tracking several objects and tried to approach modelling eye movements during MOT using analytical models and neural networks model. We only tried to predict eye positions based on positions of objects in each frame. Our proposed linear crowding strategy explained variability of eye movements comparably well as real eye movements. This does not mean that eye movements follow presented strategy, but only that human eye movements tracking strategy can be approximated by our artificial strategy. Our strategy proposes that eye movements are influenced not only by targets, but also by distractors. Difference in tracking performance for crowding strategy over centroid strategy can be seen in trials with many distractors, where people have to deal with increased crowding. There are several drawbacks of our analytical models. Because linear crowding could be noncontinuous (because of rounding used for speeding up computation), it could predict two areas distant from each other. This behavior is not biologically plausible, because saccading from one place to another is costly (no data is processed during saccades), so people sometimes could look at place with non-optimal value instead of moving to optimum. It would be interesting to encompass time information (i.e. add information about last positions) into the model to make it more robust and hopefully it would lead to better prediction.
Another possible approach would be by modelling eye movements using bayesian probability. For each point x, y of visual field, we could assign probability $P(\vec{x}_i|\vec{x}_j)$ of changing eye movements from position $\vec{x}_i$ to position $\vec{x}_j$. Then we would be able to sample eye trajectories from this distribution. However, it is unclear how to include dots positions into probability distribution. If we simply normalize values from linear crowding strategy into probability distribution, NSS values of inferred trajectories would not be too different from original linear crowding strategy, because smoothing scanpaths with Gaussian would minimize differences.

Conclusion

In our study, psychological experiment was successfully conducted and consistency of eye trajectories during repeated presentations of same trials were studied. We have found out that if we increase number of distractors, consistency of eye trajectories will decrease. Trajectories were compared using NSS metric, which was widely described and it was presented with visualization of comparison of two trajectories. Several strategies were proposed which relate tracking strategy not only with target positions but with distractors position as well. Those strategies were compared and best results were obtained for linear crowding strategy. This strategy explains variability of eye movements significantly better then centroid strategy which predicts eye position from targets only. Linear crowding strategy has some limitations. Because it is an unbounded function, neural network was not able to learn this strategy properly. However when the network was trained to predict eye positions from behavioral data, it was able to predict eye positions little worse than centroid strategy (which can be learned be neural network very well). Modified version of CrowdMOT experiment was replicated to get more data for testing neural network models. In order to train networks, data was smoothed to remove artifacts in eye trajectories unrelated to dot positions. Important operation on data was permuting the inputs; it increased prediction significantly. Another contribution of our work was the development of heuristics for finding blinks in behavioral data.
We hope this study contributed to better understanding of processes behind tracking several objects and maybe it will take us one little step closer to understanding human visual system.

References

Alvarez, G. A., & Franconeri, S. L. (2007). How many objects can you track? Evidence for a resource-limited attentive tracking mechanism. *Journal of Vision*, *7*(13). doi: 10.1167/7.13.14

Alvarez, G. A., Horowitz, T. S., Arsenio, H. C., DiMase, J. S., & Wolfe, J. M. (2005). Do Multielement Visual Tracking and Visual Search Draw Continuously on the Same Visual Attention Resources? *Journal of Experimental Psychology: Human Perception and Performance*, *31*(4), 643–667. doi: 10.1037/0096-1523.31.4.643

Amari, S. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, *5*(4-5), 185–196. doi: 10.1016/0925-2312(93)90006-O

Awh, E., & Pashler, H. (2000). Evidence for split attentional foci. *Journal of Experimental Psychology: Human Perception and Performance*, *26*(2), 834–846. doi: 10.1037/0096-1523.26.2.834

Bouma, H. (1970). Interaction effects in parafoveal letter recognition. *Nature*, *226*(5241), 177–178. doi: 10.1038/226177a0

Bouma, H. (1973). Visual interference in the parafoveal recognition of initial and final letters of words. *Vision Research*, *13*(4), 767–782. doi: 10.1016/0042-6989(73)90041-2

Brainard, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision*, *10*(4), 433–436. doi: 10.1163/156856897X00357

Broadbent, D. E. (1952). Listening to one of two synchronous messages. *Journal of Experimental Psychology*, *44*(1), 51–55. doi: 10.1037/h0056491

Bronkhorst, A. W. (2000). The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions. *Acta Acustica United With Acustica*, *86*(1), 117–128.

Cavanagh, P., & Alvarez, G. A. (2005). Tracking multiple targets with multifocal attention. *Trends in Cognitive Sciences*, *9*(7), 349–354. doi: 10.1016/j.tics.2005.05.009

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with 2 ears. *Journal of the Acoustical Society of America*, *25*(5), 975–979. doi: 10.1121/1.1907229

Clauss, M., Bayerl, P., & Neumann, H. (2004). A statistical measure for evaluating regions-of-interest based attention algorithms. *Pattern Recognition*, 383–390. doi: 10.1007/978-3-540-28649-3_47

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, *74*(368), 829–836. doi: 10.2307/2286407

Cornsweet, T. N. (1962). The staircase-method in psychophysics. *The American Journal of Psychology*, *75*(3), 485–491. Retrieved from `www.jstor.org/stable/10.2307/1419876`

Demuth, H., & Beale, M. (1992). *Neural Network Toolbox For Use with MATLAB*. The MathWorks, Inc.

Deutsch, J. A., & Deutsch, D. (1963). Attention: Some Theoretical Considerations. *Psychological Review*, *70*(1), 80–90. doi: 10.1037/h0039515

Dorr, M., Martinetz, T., Gegenfurtner, K. R., & Barth, E. (2010). Variability

of eye movements when viewing dynamic natural scenes. *Journal of vision*, *10*(10). doi: 10.1167/10.10.28

Driver, J. (2001). A selective review of selective attention research from the past century. *British Journal of Psychology*, *92 Part 1*(1), 53–78. doi: 10.1348/000712601162103

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (2nd ed. ed., Vol. 2; R. O. Duda, P. E. Hart, & D. G. Stork, Eds.) (No. 6). New York: J. Wiley. doi: 10.1038/npp.2011.9

Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology: General*, *113*(4), 501–517. doi: 10.1037/0096-3445.113.4.501

Eriksen, C. W., & St James, J. D. (1986). Visual attention within and around the field of focal attention: a zoom lens model. *Perception and Psychophysics*, *40*(4), 225–240. doi: 10.3758/BF03211502

Fehd, H. M. (2009). *Eye movement strategies during attentional tracking*. Unpublished doctoral dissertation, Vanderbilt University.

Fehd, H. M., & Seiffert, A. E. (2008). Eye movements during multiple object tracking: where do participants look? *Cognition*, *108*(1), 201–209. doi: 10.1016/j.cognition.2007.11.008

Fehd, H. M., & Seiffert, A. E. (2010). Looking at the center of the targets helps multiple object tracking. *Journal of Vision*, *10*(4), 19.1–13. doi: 10.1167/10.4.19

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *Neural Networks, IEEE Transactions on*, *5*(6), 989–993. doi: 10.1109/72.329697

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed. ed., Vol. 13; J. Griffin, Ed.) (No. 4). Upper Saddle River: Prentice Hall. doi: 10.1017/S0269888998214044

He, S., Cavanagh, P., & Intriligator, J. (1996). Attentional resolution and the locus of visual awareness. *Nature*, *383*(6598), 334–337. doi: 10.1038/383334a0

Intriligator, J., & Cavanagh, P. (2001). The Spatial Resolution of Visual Attention. *Cognitive Psychology*, *43*(3), 171–216. doi: 10.1006/cogp.2001.0755

Kleiner, M., Brainard, D. H., & Pelli, D. G. (2007). What's new in Psychtoolbox. *Perception*, *36*(ECVP Abstract Supplement).

Kooi, F. L., Toet, A., Tripathy, S. P., & Levi, D. M. (1994). The effect of similarity and duration on spatial interaction in peripheral vision. *Spatial Vision*, *8*(2), 255–279. Retrieved from `http://www.ncbi.nlm.nih.gov/pubmed/7993878`

Landry, S. J., Sheridan, T. B., & Yufik, Y. M. (2001). A methodology for studying cognitive groupings in a target-tracking task. *IEEE Transactions on Intelligent Transportation Systems*, *2*(2), 92–100. doi: 10.1109/6979.928720

Levi, D. M. (2008). Crowding–an essential bottleneck for object recognition: a mini-review. *Vision Research*, *48*(5), 635–654. doi: 10.1016/j.visres.2007.12.009

McConkie, G. W. (1981). Evaluating and reporting data quality in eye movement research. *Behavior Research Methods*, *13*(2), 97–106. doi: 10.3758/

BF03207916

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. doi: 10.1007/BF02478259

Moray, N. (1959). Attention in dichotic listening: Affective cues and the influence of instructions. *The Quarterly Journal of Experimental Psychology*, *11*(1), 56–60. doi: 10.1080/17470215908416289

Navon, D., & Miller, J. (2002). Queuing or sharing? A critical evaluation of the single-bottleneck notion. *Cognitive Psychology*, *44*(3), 193–251. doi: 10.1006/cogp.2001.0767

Norman, D. A. (1968). Toward a theory of memory and attention. *Psychological Review*, *75*(6), 522. doi: 10.1037/h0026699

Noton, D., & Stark, L. (1971). Scanpaths in eye movements during pattern perception. *Science*, *171*(968), 308–311. doi: 10.1126/science.171.3968.308

Oksama, L., & Hyönä, J. (2004). Is multiple object tracking carried out automatically by an early vision mechanism independent of higher-order cognition? An individual difference approach. *Visual Cognition*, *11*(5), 631–671. doi: 10.1080/13506280344000473

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: transforming numbers into movies. *Spatial Vision*, *10*(4), 437–442. Retrieved from `http://www.ncbi.nlm.nih.gov/pubmed/9176953`

Pelli, D. G., Palomares, M., & Majaj, N. J. (2004). Crowding is unlike ordinary masking: Distinguishing feature integration from detection. *Journal of Vision*, *4*(12), 1136–1169. doi: 10.1167/4.12.12

Pelli, D. G., Tillman, K. A., Freeman, J., Su, M., Berger, T. D., & Majaj, N. J. (2007). Crowding and eccentricity determine reading rate. *Journal of Vision*, *7*(2), 20.1–36. doi: 10.1167/7.2.20

Peters, R. J., Iyer, A., Itti, L., & Koch, C. (2005). Components of bottom-up gaze allocation in natural images. *Vision Research*, *45*(18), 2397–2416. doi: 10.1016/j.visres.2005.03.019

Posner, M. I., Snyder, C. R., & Davidson, B. J. (1980). Attention and the detection of signals. *Journal of Experimental Psychology*, *109*(2), 160–174. doi: 10.1037/0096-3445.109.2.160

Pylyshyn, Z. W., & Storm, R. W. (1988). Tracking multiple independent targets: evidence for a parallel tracking mechanism. *Spatial Vision*, *3*(3), 179–197. doi: 10.1163/156856888X00122

Rajashekar, U., Cormack, L. K., & Bovik, A. C. (2004). Point of gaze analysis reveals visual search strategies. *Human Vision and Electronic Imaging IX*, *5292*(1), 296–306. doi: 10.1117/12.537118

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, *124*(3), 372–422. doi: 10.1037/0033-2909.124.3.372

RDC Team, R. (2012). *R: A Language and Environment for Statistical Computing* (Vol. 1) (No. 2.11.1). Vienna, Austria: R Foundation for Statistical Computing. Retrieved from `http://www.r-project.org`

Rojas, R. (1996). *Neural Networks A Systemic Introduction* (2nd ed. ed.). New York: Springer-Verlag.

Sanders, A. F. (1967). Information processing in the functional visual field. *Ned-*

erlands Tijdschrift voor de Psychologie en haar Grensgebieden, *22*(3), 137–149. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/6043661

Santella, A., & DeCarlo, D. (2004). Robust clustering of eye movement recordings for quantification of visual interest. In *Proceedings of the eye tracking research & applications symposium on eye tracking research & applications - etra'2004* (pp. 27–34). New York, New York, USA: ACM Press. doi: 10.1145/968363.968368

Savitzky, A., & Golay, M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, *36*(8), 1627–1639. doi: 10.1021/ac60214a047

Scholl, B. J., Pylyshyn, Z. W., & Feldman, J. (2001). What is a visual object? Evidence from target merging in multiple object tracking. *Cognition*, *80*(1-2), 159–177. doi: 10.1016/S0010-0277(00)00157-8

Styles, E. A. (2006). *The psychology of attention* (2nd ed. ed.). New York: Psychology Press.

Tatler, B. W., Baddeley, R. J., & Gilchrist, I. D. (2005). Visual correlates of fixation selection: effects of scale and time. *Vision research*, *45*(5), 643–59. doi: 10.1016/j.visres.2004.09.017

Treisman, A. M. (1960, October). Contextual cues in selective listening. *Quarterly Journal of Experimental Psychology*, *12*(4), 242–248. doi: 10.1080/17470216008416732

Trenn, S. (2008). Multilayer perceptrons: approximation order and necessary number of hidden units. *IEEE Transactions on Neural Networks*, *19*(5), 836–844. doi: 10.1109/TNN.2007.912306

Verstraten, F. A. J., Cavanagh, P., & Labianca, A. T. (2000). Limits of attentive tracking reveal temporal properties of attention. *Vision Research*, *40*(26), 3651–3664. doi: 10.1016/S0042-6989(00)00213-3

Šíma, J., & Neruda, R. (1996). *Teoretické otázky neuronových sítí*. Praha: Matfyzpress.

Welford, A. T. (1952). The psychological refractory period and the timing of high speed performance: A review and a theory. *British Journal of Psychology. General Section*, *43*(1), 2–19. doi: 10.1111/j.2044-8295.1952.tb00322.x

Yantis, S. (1992). Multielement visual tracking: attention and perceptual organization. *Cognitive Psychology*, *24*(3), 295–340. doi: 10.1016/0010-0285(92)90010-Y

Zelinsky, G. J., & Neider, M. B. (2008). An eye movement analysis of multiple object tracking in a realistic environment. *Visual Cognition*, *16*(5), 553–566. doi: 10.1080/13506280802000752

Zelinsky, G. J., & Todor, A. (2010). The role of "rescue saccades" in tracking objects through occlusions. *Journal of Vision*, *10*(14), 1–13. doi: 10.1167/10.14.29

List of Tables

List of Abbreviations

- ANN – Artificial neural networks
- L-M – Levenberg-Marquardt algorithm
- MLP – Multi-Layer Perceptron
- MOT – Multiple object tracking
- NSS – Normalized scanpath saliency
- RT – Reaction time

Attachment 1 – Czech translation of crucial terms

There is a lack of Czech literature concerning topics covering distributed attention, crowding and eye movements. We propose translation of several crucial terms and we hope that with increasing interest in modelling of cognitive processes, some consensus in terminology can be achieved.

- Multiple object tracking – sledování více objektů
- Crowding – stísnění
- Scanpath – zkoumaná cesta
- Normalized scanpath saliency – salience normalizované zkoumané cesty

Attachment 2 – Description of the source code and the data

We'd like to briefly introduce key parts of our source code, more detailed documentation can be found on attached cd.
We programmed our experiment in MATLAB with Psychtoolbox-3 and Neural Network toolbox (version 7.0.3) installed. Both toolboxes are necessary for correct functioning of the program.
Source code could be divided into three main parts:

1. Experiment – code used for preparing data for the experiment (generating dot trajectories and experiment protocols), for main presentation part of the experiment and for parsing measured eye movements. This part require only Psychtoolbox-3 installed.

2. NSS computation and strategies – code used for computing NSS values and for generating analytical strategies. This part requires only Psychtoolbox-3 installed.

3. Machine learning – code used for training neural networks. This part requires both Psychtoolbox-3 and Neural network toolbox installed.

Experiment can be run without eye tracker, but it will serve only as illustration of real experiment, because eye data were crucial in our research. Each part has several main functions.

1. There are three main functions in Experiment part

   - `PrepareExperiment` - function which creates experimental protocols and generates trajectories. It should be run first, if we want to prepare data for new experiment
   - `StartExperiment` - main method for administrating experiment, it visualizes trajectories and collects responses from participant
   - `SaveAsMat` - function which parses data from experiment and saves them as .mat files which are used for computation of NSS values and for machine learning.

2. There is one main function in NSS computation part

   - `ComputeAllNSS` - Prepare fixation map for all trials (fixation maps are in source code denoted as scanpath space) and computes NSS values for trials. It also computes analytical strategies and compares them using NSS metric. All results saves into the file.

3. There is one main function in Machine learning part

   - `Start` - Prepares training data for neural networks using several operations as mentioned in text. Trains neural networks on this data and validates it using datasets from CrowdMOT and CrowdMOT02 experiment.

Code in each part uses shared configuration for easier modification. There is more detailed documentation on the attached cd. Each function is documented and this documentation can be viewed using standard MATLAB function `help functionname` or if we want to see all description of all functions in current directory we can type `help(cd)` which loads information from Contents.m file. All experiment data can be found on the attached cd. There are several main .mat files which contains all experiment results.

- `eyeData.mat` – contains parsed eye trajectories from experiment. All eye trajectories have normalized length to 7400ms and do not contain blinks.
- `trackData.mat` – contains all dot trajectories from trials used in experiment.
- `response.mat` – contains participants' responses from experiment
- `nsspath.mat` – contains NSS values for all trials
- `nsspath_strategy.mat` – contains NSS values for all strategies

Trained neural networks and inputs which were used for training are on the cd as well.